

Prozessgebundene Berechnungsbaugruppen - Ein Ansatz zur Lösung komplexer  
Entscheidungs- und Berechnungsabläufe

Von der Fakultät Maschinenwesen

der

Technischen Universität Dresden

zur

Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommene Dissertation

Dipl.-Ing. Denis Polyakov

geb. am 15. Dezember 1983 in Klimovsk (Russland)

Tag der Einreichung: 22.08.2017

Tag der Verteidigung: 26.03.2018

Gutachter: Prof. Dr.-Ing. Berthold Schlecht (TU Dresden)

Prof. Dr.-Ing. habil. Ralph Stelzer (TU Dresden)

Vorsitzender der Promotionskommission: Herr Prof. Dr.-Ing. habil. Thorsten Schmidt (TU Dresden)

## Vorwort und Dank

Die vorliegende Arbeit entstand während meiner Tätigkeit als Entwicklungsingenieur in Dresden. Sie beschäftigt sich mit der prozessorientierten Sicht auf die Berechnung des Verhaltens von Maschinenelementen und Baugruppen unter betrieblichen Belastungen. Als Beitrag zur konstruktionsbegleitenden Simulation adressiert sie insbesondere die digitalen Voraussetzungen für eine durchgängige Integration von Berechnungen in den modellbasierten Produktentwicklungsprozess.

Mein Dank gilt vor allem Herrn Prof. Dr.-Ing. Berthold Schlecht für das entgegengebrachte Vertrauen und die vielfältigen Möglichkeiten, die Ressourcen seines Institutes zu nutzen. Durch seine umfassende Unterstützung und Betreuung erhielt ich entscheidende Impulse für das Gelingen der Arbeit.

Herrn Prof. Dr.-Ing. habil. Ralph Stelzer möchte ich danken für seine kritische Durchsicht der Arbeit, wertvolle Anregungen und die Übernahme des Zweitgutachtens.

Mein Dank gilt weiterhin Herrn Prof. Dr.-Ing. habil. Thorsten Schmidt, der sich bereit erklärt hat, den Vorsitz der Prüfungs- und Promotionskommission zu übernehmen.

Besonderen Dank schulde ich meinem langjährigen Mentor Herrn Prof. h.c. Dr.-Ing. Willi Gründer für die zahlreichen fachlichen Diskussionen und seine wertvollen Ratschläge. Außerdem bin ich ihm sehr dankbar für die bei meinem Arbeitgeber TEDATA GmbH geschaffenen Voraussetzungen für den erfolgreichen Abschluss meiner Arbeit.

Für die aufgeschlossene Atmosphäre und die vielen hilfreichen Diskussionen möchte ich mich bei den Mitarbeitern der TEDATA GmbH und der Driveconcepts GmbH bedanken, besonders bei Herrn Dipl.-Ing. Konrad Riedel für die fruchtbaren Auseinandersetzungen sowie bei Herrn Dr.-Ing. Tobias Schulze und Herrn Dipl.-Ing. Jan Hebig für die stete Bereitschaft, mir mit Rat und Tat zur Seite zu stehen.

Zu außerordentlichem Dank bin ich meiner Frau Elena und meiner Tochter Alexandra verpflichtet. Sie waren immer auf meiner Seite und haben mir stets den nötigen Rückhalt gegeben, obwohl sie während meiner Promotionszeit auf viele gemeinsame Stunden verzichten mussten. Weiterhin danke ich meinem Papa Serguei für die wegweisenden Diskussionen und meinem Bruder Dmitry für die immer zum richtigen Zeitpunkt gefundenen aufmunternden Worte.

Bochum, 18.06.2018

Denis Polyakov



# Inhaltsverzeichnis

1	Einleitung .....	4
1.1	Hintergrund und Motivation .....	4
1.2	Zielsetzung der Arbeit .....	6
1.3	Lösungsweg.....	7
1.4	Durchgeführte Untersuchungen und Arbeiten.....	8
1.5	Aufbau der Arbeit .....	9
2	Stand der Technik und Forschung.....	11
2.1	Produktentwicklung.....	11
2.1.1	Interdisziplinarität.....	13
2.1.2	Prozesskomplexität .....	14
2.1.3	Informationsmanagement und Dokumentation.....	15
2.1.4	Wissensmanagement .....	17
2.1.5	Analytische Berechnung in der Konstruktion.....	18
2.1.6	IT-Unterstützung in der Produktentwicklung .....	19
2.2	Konstruktionsmethoden in Maschinenbau .....	21
2.2.1	Vorgehensmodelle .....	21
2.2.2	Das V-Modell der VDI-2206.....	24
2.2.3	Modularisierung (Prozessbausteine) .....	26
2.2.4	Prozessorientierte Betrachtung der Konstruktion.....	27
2.3	Modellbildung.....	31
2.3.1	Modelldefinition .....	31
2.3.2	Modellbildung in Konstruktion .....	32
2.3.3	Prozessmodelle.....	32
2.3.4	Grundlagen der Prozessmodellierung .....	33
2.4	Prozessautomatisierung.....	36
2.4.1	Petri-Netze.....	37

2.4.2	UML - Aktivitätsdiagramme .....	39
2.4.3	Ereignisgesteuerte Prozessketten.....	42
2.4.4	Business Process Management und Notation (BPMN).....	43
2.5	Systems Engineering (SE).....	46
2.5.1	Modellbasiertes Systems Engineering (MBSE).....	48
2.5.2	System Modeling Language (SysML) .....	49
2.5.3	Einordnung der analytischen Berechnungen in MBSE .....	56
2.5.4	Modellbasierte vs. modellgetriebene Entwicklung.....	59
2.6	Getriebeentwicklungsprozesses .....	61
2.6.1	Berechnung von Zahnradgetrieben .....	61
2.6.2	EDV-Werkzeuge für Auslegungen und Parameterstudien .....	65
3	Prozessgebundene Berechnungsbaugruppen.....	66
3.1	Theoretische Untersuchungen .....	68
3.1.1	Anforderungen .....	68
3.1.2	Darstellung von Anforderungen .....	70
3.1.3	Formalisierung von Anforderungen .....	71
3.1.4	Produktmerkmale .....	72
3.1.5	Klassifizierung von Produktmerkmalen.....	74
3.1.6	Merkmalseigenschaften eines Objektes .....	77
3.1.7	Einheitensystem von Merkmalen.....	79
3.2	Ermittlung von Merkmalswerten .....	80
3.2.1	Funktionsbaustein als elementar funktionales Prozesselement.....	82
3.2.2	Datenaustausch zwischen den Funktionsbausteinen.....	83
3.2.3	Modellierung von Berechnungsprozessen.....	86
3.2.4	Formalisierung der Funktionsbausteine.....	87
3.2.5	Strukturierung der Merkmale und Funktionsbausteine .....	90
3.3	Wiederverwendung von Konzepten .....	93
3.3.1	Aufbau und Nutzung von Bibliotheken.....	93

3.3.2	Sub-Prozesse .....	96
3.3.3	Fazit.....	98
3.4	Entwicklung des workfloworientierten Management-Systems .....	98
3.4.1	Design Process Management System (DPMS).....	99
3.4.2	Datenschnittstellen – Projekt-Datei .....	105
3.4.3	Definition eines Funktionsbausteines.....	107
3.4.4	Modellierung von Berechnungsprozessen.....	109
3.4.5	Ausführung von Prozessen und Sub-Prozessen.....	115
3.4.6	Generierung und Ausführung der Applikationen.....	120
3.4.7	Fazit.....	121
4	Anwendung des Ansatzes am Beispiel Getriebeberechnung.....	123
4.1	Theoretische Grundlagen des Berechnungsbeispiels.....	123
4.1.1	Verzahnungsberechnung .....	123
4.2	Umsetzung mit dem Prozess-Designer.....	129
4.2.1	Beurteilung der Ergebnisse .....	134
5	Zusammenfassung und Ausblick .....	136
5.1	Zusammenfassung .....	136
5.2	Ausblick.....	137
6	Literaturverzeichnis.....	139
7	Abbildungsverzeichnis .....	150
8	Tabellenverzeichnis.....	155
9	Formel- und Abkürzungsverzeichnis .....	156
10	Anhang A.....	157
11	Anhang B.....	198

# 1 Einleitung

## 1.1 Hintergrund und Motivation

Die Produktentwicklung unterliegt unter Nachhaltigkeitsaspekten, den Bedingungen erweiterter Märkte und zunehmender Regulierungsdichte immer mehr äußeren Einflüssen. Diese führen in ihrer Vielfalt und Komplexität einerseits zu einer wachsenden zeitlichen Belastung, vor allem aber zu einer höheren Aufgabendichte. Wenn diese Anforderungen unter dem Eindruck anspruchsvoller Produktivitätsziele dennoch bewältigt werden sollen, sind eine weitergehende Digitalisierung und Automatisierung der Arbeitsvorgänge und ihrer Kommunikation untereinander unausweichlich. Voraussetzung hierfür sind allerdings prozessgebundene Abläufe, die in ihrer Differenziertheit transparent und modularisierbar sind und weder Halt vor Grenzen, noch vor neuen Fragestellungen machen. Prozessorientierung bedeutet dabei eine horizontal durch die Organisation verlaufende Vorgangskette (siehe Abbildung 1)

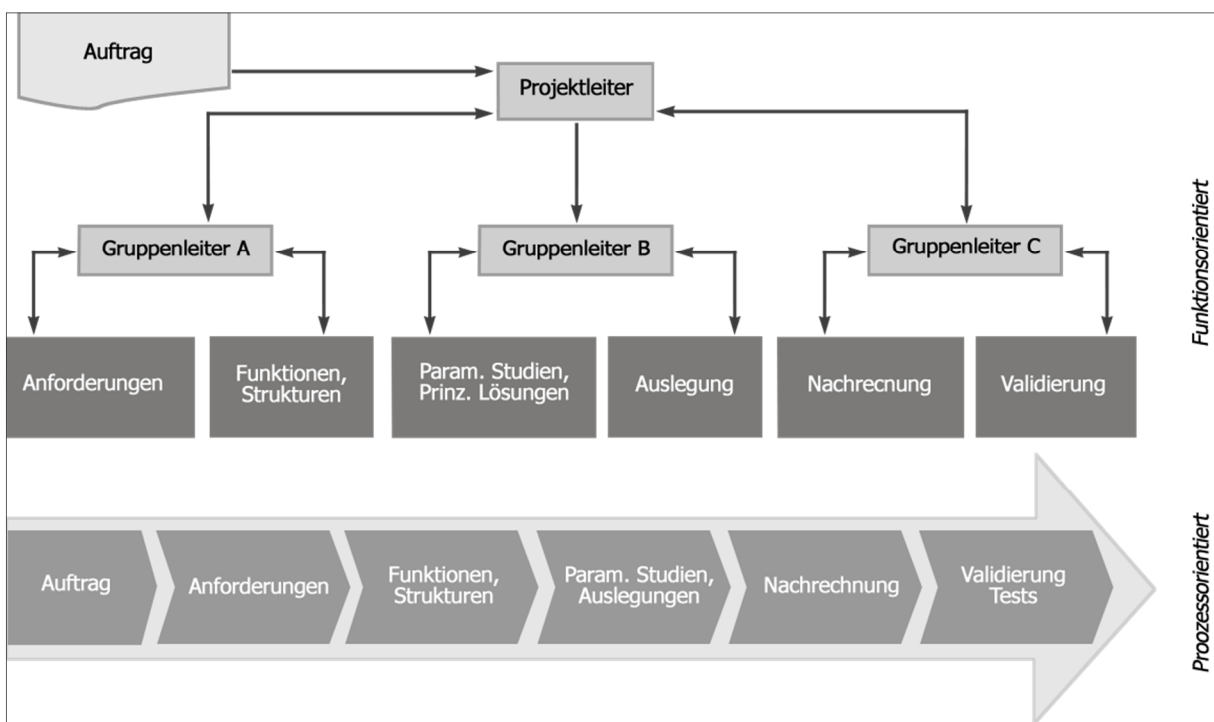


Abbildung 1: Funktionsorientierte vs. prozessorientierte Produktentwicklung

Im Gegensatz hierzu steht die funktionsbezogene Vorgangsbearbeitung, die hierarchisch stark strukturiert ist. Diese Funktionsorientierung ist immer noch in vielen großen Unternehmen zu beobachten und meistens auf die historische Entwicklung des Unternehmens zurückzuführen. „Gewachsene“ Funktionsorientierungen stellen heute in vielen Organisationen Reorganisationshindernisse dar, da sie ein hohes Maß an Planung und Koordination erfordern (vgl. Vorest AG, 2015, Xiang, 2015, Helming, 2016). Außerdem wird die Integration neuer Abläufe durch die Funktionsorientierung so erschwert, dass eine flexible Reaktion auf dynamisch wechselnde

Anforderungen kaum möglich ist. Als Folge wird jeder Schritt über Abteilungsgrenzen als eine potenzielle Störung wahrgenommen (Vorest AG, 2015, BMI, 2016). Die Prozessorientierung kennt dagegen, zumal wenn sie durch eine fortschreitende Digitalisierung getragen wird, Unternehmens- und Abteilungsgrenzen nicht als Barrieren, sondern als Kontrollpunkte, da die Verantwortlichkeiten zuallererst an den Prozess und nicht an die Funktion gebunden sind. Sehr gute Voraussetzungen für die Realisierung solcher Prozessarchitekturen bilden leicht handhabbare und zu konfigurierende Workflows, da sie die Grundlage für die zuverlässige und flexible Integration einer Vielzahl von Aufgaben und Kommunikationsmechanismen bilden.

Das gilt auch für den technischen Bereich der Produktentwicklung. Die Komponenten der komplexen maschinenbaulichen Erzeugnisse werden oft immer noch losgelöst vom ganzen System ausgelegt und berechnet, was einen erheblichen Anpassungsaufwand der Komponenten und Baugruppen zur Folge haben kann. Außerdem stellt sich in vielen Ketten von Berechnungen das Problem der Konsistenz, wenn in Teilberechnungen Veränderungen vorgenommen werden. Sofort müssen die Auswirkungen in allen anderen Teilberechnungen überprüft werden. Ein auftretender Iterationsbedarf kann dabei beträchtlich sein, aus diesem Grund werden regelmäßig problembezogene Verfahrensketten entwickelt und programmiert, die den Optimierungsprozess ganz oder in Teilen automatisieren helfen. Diese Vorgehensweise stößt immer dann an ihre Grenzen, wenn neben der Anzahl der Dimensionsvarianten auch das Aufkommen an Form- und Funktionsvarianten wächst. Starre und monolithisch verfasste Programme, die zudem mit alten Technologien entwickelt wurden, erweisen sich dann als wenig flexibel und können nur mit großem Aufwand an die neuen Aufgabenstellungen angepasst werden. Außerdem können zwischen der ersten Inaugenscheinnahme einer Aufgabenstellung und Fertigstellung der Entwicklung viele Monate liegen. Meist haben mehrere Konstrukteure an dem Prozess mitgewirkt, es wurden Unmengen von Entscheidungen getroffen und ebenso viele Berechnungen durchgeführt. Am Ende liegen Ergebnisse vor, die zwar ausnahmslos dokumentiert sind, doch wie sie zustande kamen, bleibt oft im Verborgenen. Die Ursachen hierfür liegen einerseits in unverbindlichen Regelwerken, andererseits in den nicht dokumentierten Kreativitätsschüben, welche eine Fülle von detaillierten Untersuchungen und Iterationen verursachen, deren Dokumentation meistens aus den Zeitgründen unterlassen wird und die deshalb ebenso einen negativen Einfluss auf die Nachvollziehbarkeit ausüben. Zusammenfassend lassen sich die Probleme bei der Auslegung und Berechnung von Maschinenelementen und Maschinen-Baugruppen wie folgt beschreiben:

- zu große Komplexität sowie oft archaischer Aufbau von Berechnungsabläufen bzw. Berechnungsschemata,
- mangelnde Transparenz der Abläufe und damit deren aufwendige Konsistenzprüfung sowie
- zeit- und ressourcenaufwendiges Änderungswesen der Abläufe.

## 1.2 Zielsetzung der Arbeit

Die vorliegende Arbeit will Wege zur Prozessintegration eines der wichtigsten Teilprozesse der Produktentwicklung aufzeigen, der Berechnung der in einem Erzeugnis verbauten Maschinenelemente und Baugruppen. Deren Merkmale und Auslegungskriterien unterliegen bezüglich der Sicherheit, der Lebensdauer und der Kosten, aber auch im Hinblick auf ihren Einsatzort, ihre Einsatzbedingungen und die einzelnen Lebensphasen einer hochdynamischen Parametrik und Methodik.

Mit den hier vorgestellten Erkenntnissen und Entwicklungen soll ein Beitrag dazu geleistet werden, den arbeitsteiligen und oft fragmentierten Entwicklungsprozess und die dabei zahlreich ablaufenden einzelnen, auch parallel ablaufenden Schritte und interdisziplinären Aktivitäten transparenter, schneller und qualitätsorientierter zu gestalten. Die Umsetzung prozessorientierter Vorgehensweisen bei der Berechnung der zu Baugruppen verbundenen Maschinenelemente wird im Rahmen der vorliegenden Arbeit als „Konzept der Prozessgebundenen Berechnungsbaugruppen“ bezeichnet.

Die Praktikabilität des Konzeptes wird am Beispiel einer Zahnradgetriebeberechnung nachgewiesen. Die Berechnung von Getrieben eignet sich hierfür besonders gut, da diese einen komplexen Prozess mit starker Vernetzung und gegenseitiger Einflussnahme der Teilprozesse aufeinander repräsentiert. Ein besonderes Augenmerk soll dabei auf die frühen Phasen der Konstruktion bzw. Berechnung eines Getriebes gelegt werden.

Ausgehend von den genannten Problemen kann die Zielsetzung für die vorliegende Arbeit wie folgt zusammengefasst werden:

- Entwicklung einer Methode für den systematischen und dennoch flexiblen Aufbau und die automatische Ausführung von Berechnungsprozessen für Baugruppen-berechnungen.

Diese Methode soll folgendes ermöglichen:

- Definition der Randbedingungen einer Berechnung,
- Definition und Dokumentation des Berechnungsablaufes (Struktur der Berechnung) und
- Interpretationshilfe für die gewonnenen Ergebnisse.

### 1.3 Lösungsweg

In Bezug auf die Ermittlung der Parameter einer mechanischen Konstruktion und der dazugehörigen Teil-Konstruktionen (Baugruppen) kann man sich einen Berechnungsprozess als einen iterativen Durchlauf eines gerichteten Graphen vorstellen, wobei seine Knoten die einzelnen Teilberechnungen und seine Kanten die Folgerichtung repräsentieren. Die Kanten dieses Graphen stellen dabei nicht nur die statisch bestimmte Abfolge, sondern auch den dynamischen Richtungswechsel dar, und zwar abhängig von den in den entsprechenden Knoten oder Gruppen von Knoten gewonnenen Ergebnissen.

Ein Knoten kann dabei sowohl eine elementare Berechnung beinhalten, basierend auf einer Formel aus einer Norm oder Richtlinie, als auch komplexe mathematische Sachverhalte wie die Lösung eines Differentialgleichungssystems, das einen Spannungszustand beschreibt. Außerdem können mehrere elementare Berechnungen zu einem zusammengesetzten Knoten (Teillösung) zusammengefasst werden.

Die bekannten Parameter eines Berechnungsablaufes sind dem Startknoten zugeordnet. Dieser wird nicht ausgeführt, sondern dient lediglich als Daten-Container. Analog gibt es auch einen Knoten, den sogenannten End-Knoten (siehe Abbildung 2), welcher die Ergebnissammlung übernimmt und gleichzeitig das Ende des Berechnungsablaufes kennzeichnet.

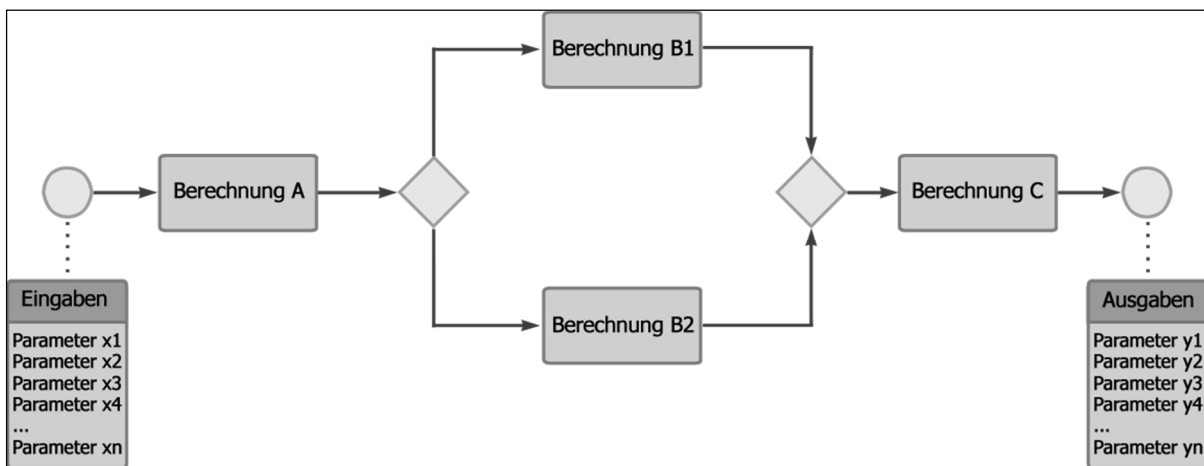


Abbildung 2: Schematischer Ablauf einer Berechnung

Ein solchermaßen aufgebauter gerichteter Graph repräsentiert somit das mathematische Modell eines Berechnungsprozesses. Für standardisierte bzw. genormte Berechnungen, zum Beispiel für die Zahnfußfestigkeitsberechnung nach DIN 3990-3:1987-12 sind die Graph-Strukturen vordefiniert und können nicht geändert werden, d. h. diese sind als Invariante einer Lösung zu verstehen (siehe Abbildung 3).

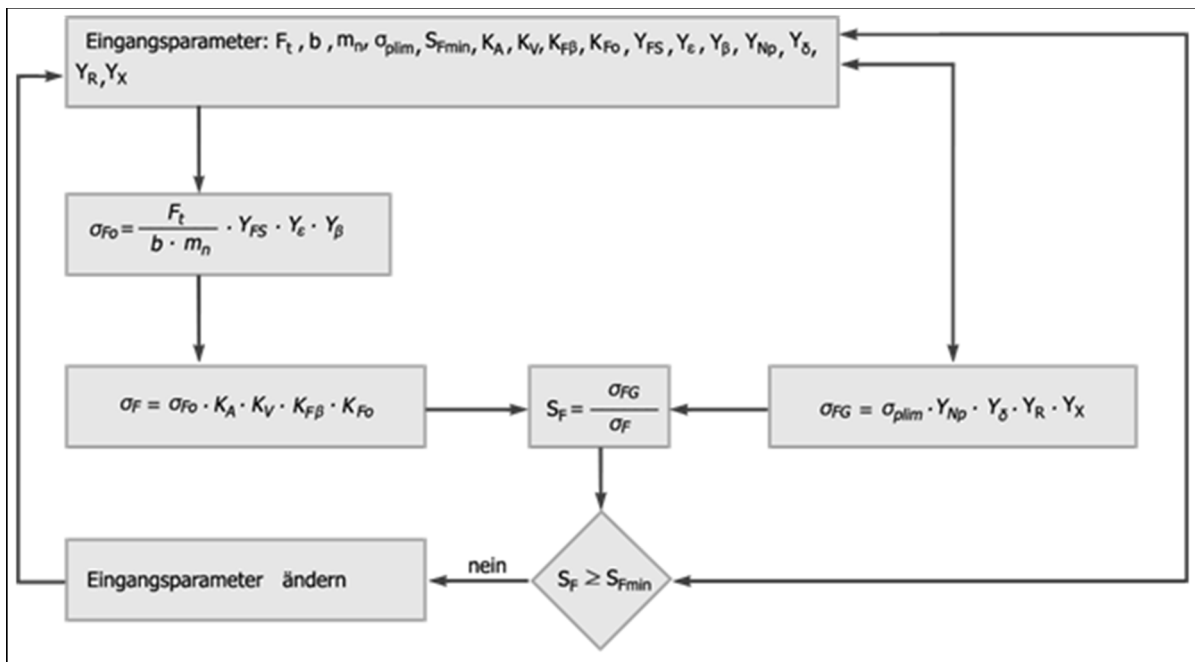


Abbildung 3: Allgemeiner Ablauf der Fußfestigkeitsberechnung nach DIN 3990:1987

Die Prozessverwaltung erfolgt in einem im Rahmen der vorliegenden Arbeit prototypisch entwickelten Workflow-Autoren-System, welches die Modellierung und Ausführung bzw. Simulation der Prozesse ermöglichen soll. Bei einer manuellen Konfiguration der Berechnungsabläufe können eigene Teillösungen (Sub-Prozesse) im System abgelegt und in andere Abläufe integriert werden. Abhängig von der neuen Aufgabenstellung können sie weiterhin geändert bzw. erweitert werden, so dass daraus völlig neue Abläufe mit anderen Knoten und Kanten entstehen.

Dieser Ansatz zur Darstellung einer Berechnung von Maschinenelementen oder Baugruppen in Form eines gerichteten Graphen zusammen mit der dazu entwickelten Semantik und der Syntax, ermöglichen eine eindeutige Formalisierung des Berechnungsablaufes, welche zwingend für die Prozessautomatisierung notwendig ist. Der bei der Modellierung entstandene Graph, inklusive aller Zwischenschritte, dokumentiert die Entstehung der technologischen Lösung und schafft gleichzeitig die Grundlage für nachfolgende Optimierungen und Verbesserungen der Prozesse.

## 1.4 Durchgeführte Untersuchungen und Arbeiten

Auf Basis der Literaturrecherche wurden die grundlegenden Vorgehensweisen und Methoden für die Entwicklung einer neuen Methode zur systematischen, prozessorientierten Organisation und Optimierung von Konstruktionsprozessen, insbesondere im Bereich der Berechnung von Maschinenelementen und Baugruppen, identifiziert. Die Analyse des Standes der Technik wurde multidisziplinär durchgeführt, indem nicht nur das Gebiet des methodischen Konstruierens, sondern auch solche Fachgebiete wie die Produktentwicklung, das Systems Engineering, das Business Process



Management, die Modellbildung und das Wissensmanagement erörtert wurden. Im Einzelnen wurden die Einflussfaktoren und möglichen Lösungsansätze unter folgenden Aspekten untersucht:

- Anforderungen an eine Konstruktion bzw. ein zu entwickelndes Produkt,
- die Modularisierung der Berechnungsprozesse und Festlegung der elementaren Bestandteile
- der Berechnungsprozess und die Strukturierung mit Sub-Prozessen,
- eine informationstechnische Beschreibung von Berechnungsprozessen,
- die graphische Abbildung eines Berechnungsprozesses,
- die Organisation des Ablaufs einer Berechnung und ihrer automatisierten Durchführung sowie
- die Dokumentation des Prozesses.

Die gewonnenen theoretischen Erkenntnisse führten zu einem prototypischen Design Management System, welches für ein nachhaltiges Wissensmanagement bei der Organisation der Berechnungsabläufe sorgt. Die Besonderheiten des entwickelten Workflow Management Systems liegen einerseits in einem Wissens-Repository, welche die modularen Bestandteile zur Definition der Abläufe beinhaltet sowie in einem interaktiven Prozess-Designer, mit Hilfe dessen die Abläufe graphisch, schnell und intuitiv generiert und ausgeführt werden können. Zur Veranschaulichung des im Rahmen der Arbeit entwickelten Ansatzes wurde ein Beispiel der Anwendung an einem Berechnungsprozess durchgeführt.

## 1.5 Aufbau der Arbeit

Die vorliegende Arbeit umfasst drei Hauptkapitel, sowie die Beschreibung der Konzeption und Umsetzung eines prototypischen Design Process Management Systems (DPMS) mit anschließender Zusammenfassung und dem Ausblick auf weitere Forschungs- und Entwicklungsaspekte.

Zu Beginn werden im ersten Kapitel die Ziele der Arbeit und deren Aktualität in der heutigen Produktentwicklung vorgestellt. Außerdem werden die theoretischen und praktischen Aufgabenstellungen zur Realisierung dieser Ziele identifiziert. Die Grundidee zur Lösung der gestellten Aufgaben sowie die allgemeinen Ansätze der neuen Methode schließen dieses Kapitel ab.

Im zweiten Kapitel „Stand der Technik und Forschung“ werden die Ergebnisse der Literaturrecherche beschrieben, welche die Untersuchung des aktuellen Standes in dem methodischen Konstruieren sowie die alternativen Vorgehensweisen aus andren Fachgebieten, wie das Prozess- und Wissensmanagement und die Prozessautomatisierung beinhaltet. Es werden Methoden zur graphischen Abbildung der Prozesse untersucht sowie die Grundlagen der Modellbildung.

Im dritten Kapitel „Prozessgebundene Berechnungsbaugruppen“ wird auf Basis der theoretisch gewonnenen Erkenntnisse aus dem vorherigen Kapitel die Ausarbeitung und Entwicklung der neuen Methode für die Handhabung der prozessgebundenen Berechnungen und deren Baugruppen beschrieben. Dieses Kapitel wurde in zwei Unterkapitel geteilt. Im ersten Unterkapitel erfolgt die theoretische Ausarbeitung der Ansätze und Lösungen der neuen Methode. Im zweiten Teil wird die informationstechnische Umsetzung der Entwicklung eines Design Process Management Systems (DPMS) zur rechnergestützten Organisation der Berechnungsabläufe erläutert.

Nachfolgend wird im letzten Kapitel „Anwendung des Ansatzes am Beispiel Getriebeberechnung“ die Anwendung der entwickelten Methode und des DPM Systems an einem Beispiel dargestellt.

## 2 Stand der Technik und Forschung

Bevor die Realisierung der im Rahmen dieser Arbeit geplanten Untersuchungen und Entwicklungen in Angriff genommen werden, sollen folgende Fragen geklärt werden:

- Wie komplex und umfangreich sind die Entwicklungsprozesse von maschinenbaulichen Erzeugnissen und welcher Komplexitätsgrad soll beherrscht werden?
- Welche internen Beziehungsstrukturen und welche Abhängigkeiten untereinander weisen die einzelnen Elemente einer Baugruppe auf?
- Welche Methoden und Vorgehensweisen werden heute für das Entwickeln eines maschinenbaulichen Erzeugnisses angewendet?
- Wie wird die Historie einer Produktentwicklung dokumentiert?

### 2.1 Produktentwicklung

Der Begriff „Produktentwicklung“ beinhaltet nicht nur eine als Organisationseinheit identifizierbare Unternehmensfunktion, sondern auch einen Prozess der Gestaltung von produzierbaren und funktionsfähigen Produkten (Ponn und Lindemann 2011), der sich von der Anforderungsdefinition und Produktplanung über Entwicklung bis hin zur Produktionsvorbereitung erstreckt (siehe Abbildung 4).

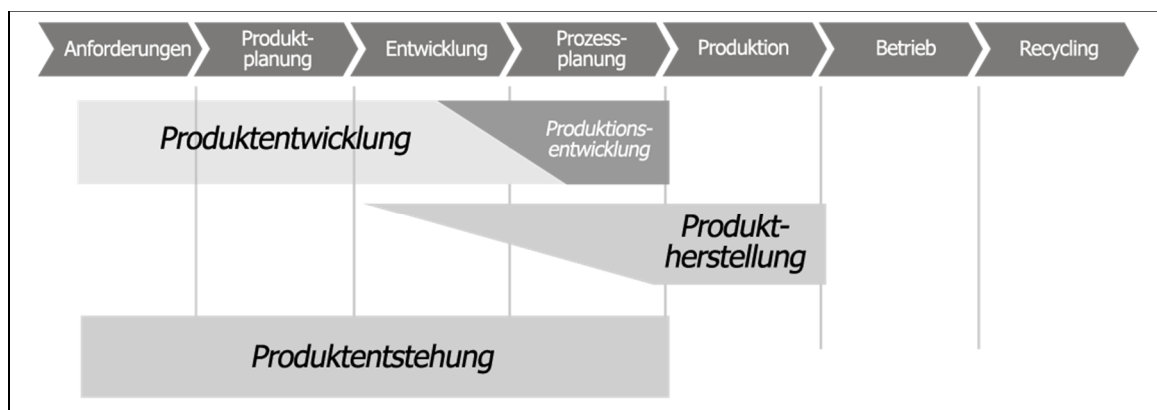


Abbildung 4: Produktentwicklung in Produktlebenszyklus (Eigner und Stelzer 2009)

Die Produktentstehung, bestehend aus der Produktentwicklung und der Produktionsentwicklung, repräsentiert dabei einen der wichtigsten Prozesse im gesamten Produktlebenszyklus, denn am Ende dieser Phase liegen alle für die Herstellung, den Betrieb und die Wiederverwendung eines Produktes notwendigen Informationen in Form von Spezifikationen, Beschreibungen, digitalen Modellen, Entwurfs- und Produktionsunterlagen sowie diejenigen aller dazugehörigen Betriebsmittel bereit (Eigner und Stelzer 2009).

Zu den wesentlichen Aspekten der Produktentwicklung gehören neben dem Produkt selbst, die eingesetzten Methoden und Werkzeuge, Prozesse zur Regelung der Abläufe und der Mensch als zentrale und verbindende Komponente (siehe Abbildung 5). Die Situation der Produktentwicklung ist dabei von einer Vielzahl von Einflüssen geprägt wie Nachhaltigkeit, Kosten- und Zeitdruck sowie vielen anderen (Ponn und Lindemann, 2011).

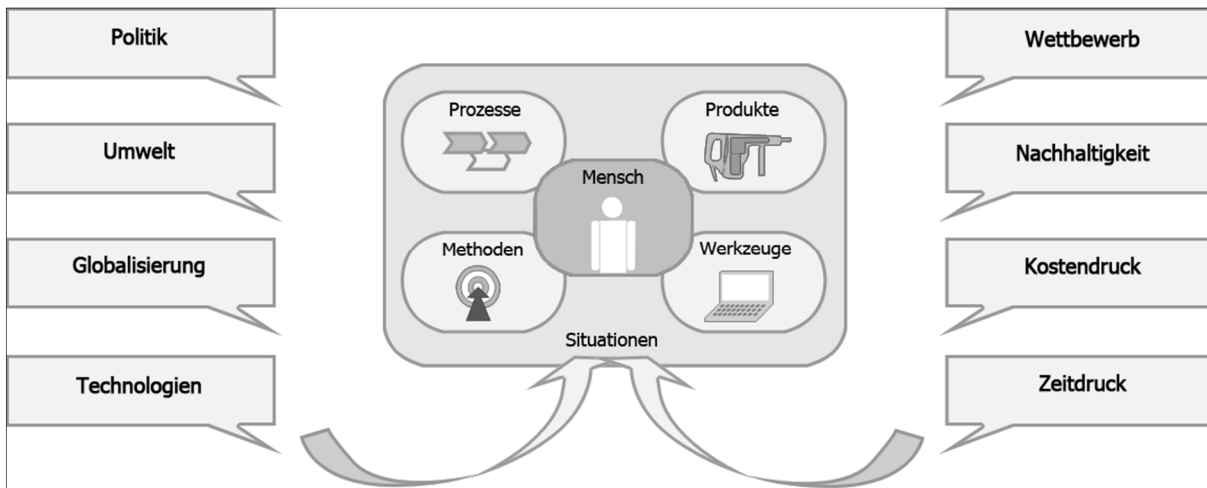


Abbildung 5: Wesentliche Aspekte der Produktentwicklung (Ponn und Lindemann, 2008)

Die Erfüllung der Kundenbedürfnisse und Anforderungen in Bezug auf die Funktion, das Design und die Kosten sind die Hauptziele jeder Produktentwicklung. Die dabei ablaufenden Prozesse regeln das Vorgehen der Entwickler sowie die Entwicklungsarbeit im Ganzen (Ponn und Lindemann, 2011).

Alle technischen Produkte bzw. maschinenbaulichen Erzeugnisse unterliegen einem Lebenszyklus. Sie werden geplant, entworfen, produziert, in Betrieb genommen, gewartet und anschließend stillgelegt und recycelt (Störrle, 2005). Die dabei angestrebte maximale Wiederverwertung des Produktes, seiner Komponenten und Werkstoffe ist ein Maß für die Nachhaltigkeit der Produktentwicklung. Auf der Abbildung 6 sind die wesentlichen Phasen des Produktlebenszyklus dargestellt.

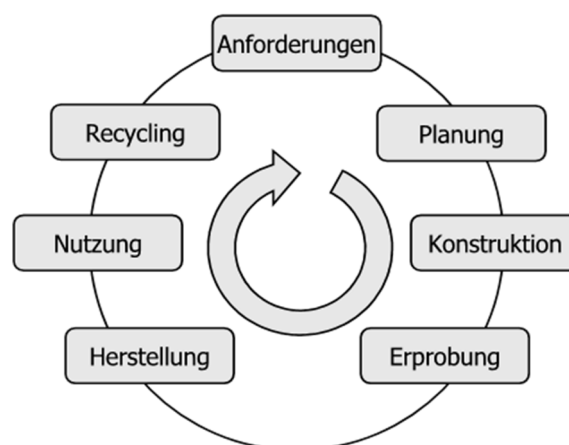


Abbildung 6: Produktlebenszyklus (adaptiert nach Eigner und Stelzer 2009)

Die Informationen aus jeder Phase fließen in die Produktentwicklung ein und machen damit die Entwicklungsprozesse noch verzweigter und komplexer. Zur Unterstützung werden planmäßige und regelbasierte Vorgehensweisen (Methoden) eingesetzt oder entwickelt. Diese bieten Hilfestellung zur Abfolge der Ausführung von bestimmten Tätigkeiten. Die Anwendung von Methoden wird durch die Nutzung entsprechender Werkzeuge erleichtert. Die Vielfalt von Werkzeugen reicht von den einfachen Formblättern und Checklisten bis zu hochkomplexen Rechen- und Simulationsprogrammen (Ponn und Lindemann, 2011). Es fehlt jedoch an den durchgängigen Entwurfsumgebungen, und die vorhandenen Insellösungen sind unzureichend integrierbar (VDI 2206:2004).

Die Informationsverarbeitung im Rahmen des Entwicklungsprozesses, bei dem die Anforderungen des Kunden unter Anwendung entsprechenden Wissens und geeigneter Methoden in die Informationen über die Gestalt und Herstellung eines Produktes überführt werden, zeigen die Wichtigkeit der Faktoren wie Information und Wissen. Besonders die früheren Phasen der Produktentwicklung, bei denen die Informationen sehr rar und unsicher sind, verleihen dem gut funktionierenden Informations- und Wissensmanagement eine hohe Bedeutung (Ponn und Lindemann, 2011).

### 2.1.1 Interdisziplinarität

Die stetig fortschreitende Virtualisierung von technischen Erzeugnissen und der damit verbundenen Entwicklungs- und Organisationsprozesse schafft gute Voraussetzungen, die Vision von der digitalen Zukunft in absehbarer Zeit wahr werden zu lassen. Die unter dem Namen „Industrie 4.0“ bekannte Forschungsrichtung beschäftigt sich bereits intensiv mit der virtuellen Abbildung des Produktionsprozesses und mit der Vernetzung ganzer Fabriken, wobei sich die Produktionsprozesse selbst aktiv steuern (eCI@ss e.V., o. D.). Letztlich kann dieses Ziel aber nur erreicht werden, wenn vollständige digitale Produktbeschreibungen vorliegen, die nicht nur das Betriebsverhalten sondern alle Phasen der Herstellung mit Hilfe der produktionsrelevanten Informationen unterstützen (Eigner, et al., 2014). Der resultierende Zwang zur Verknüpfung von realen (physischen) Produkten und Prozessen mit informationsbasierten (virtuellen) Produkten und Prozessen über offene Informationsnetze (Internet) führt zur Entstehung von sogenannten Cyber-Physikalischen Systemen (CPS). Die Entwicklung solcher Systeme bringt immer mehr die mechatronischen Aspekte in den Vordergrund. Die steigende Produktkomplexität resultiert nicht nur aus der Fülle der an den verschiedensten Kundenanforderungen ausgerichteten Produktvariationen, sondern auch aus dem ständig wachsenden Anteil an elektronischen Komponenten (vgl. ebd.).

Auch vor der Getriebeentwicklung macht die Integration der elektronischen Komponenten keinen Halt. Das Zusammenwirken mehrerer Disziplinen wie Mechanik, Elektronik und Software sorgt hier für eine Steigerung des Funktionsumfanges. Das Getriebe wird damit zu einem mechatronischen System. Die heute noch dominierenden rein mechanischen Getriebe erfahren allmählich eine Ergänzung durch eine

breite Palette von Hybrid-Produkten, in denen die Funktionen eines Verbrennungsmotors und eines Elektromotors kombiniert werden.

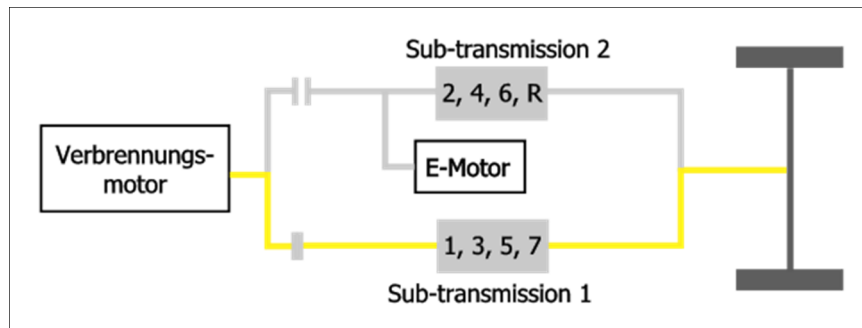


Abbildung 7: Antriebsstrang mit Hybriddrive 7HDT300 der Firma GETRAG (Blessing, 2012)

Der Elektromotor kann neben der Antriebsfunktion im Zusammenspiel mit dem Verbrennungsmotor durch das Zurückgewinnen der Bremsenergie oder der Unterstützung der Beschleunigung unterschiedliche Aufgaben übernehmen. Es sind Ersparnisse von bis zu 80 % des Kraftstoffverbrauches möglich (Blessing, 2012). Die Auslegung eines solchen Getriebes erfordert einerseits die Berücksichtigung vieler äußerer Einflüsse, andererseits aber auch die Kenntnis der gegenseitigen Wechselwirkungen zwischen der mechanischen und der elektrischen Seite des Antriebs.

Bislang erfolgt die Entwicklung oft getrennt in unterschiedlichen Domänen auf der Basis dort etablierter spezifischer Methoden, die durch disziplinen eigene Erfahrungen, Denkweisen und Begriffswelten geprägt sind. Die Erarbeitung einer optimalen Gesamtlösung für ein Produkt erfordert dagegen heute die gleichberechtigte Einbeziehung aller beteiligten Fachdisziplinen in einer Domäne übergreifenden Kooperation (VDI 2206:2004). Die Mengen an Informationen, die dabei verarbeitet und anschließend dokumentiert werden müssen, sind beachtlich. Eine Erhöhung der Komplexität der damit verbundenen Entwicklungsprozesse wird damit unvermeidbar.

### 2.1.2 Prozesskomplexität

Die Komplexität eines Systems kann in Varietät und Konnektivität unterteilt werden (Steinmeier, 1999). Dabei wird die Varietät durch die Menge und Artenvielfalt der Komponenten bestimmt, während die Konnektivität durch die Beziehungen der Elemente untereinander sowie mit der Umgebung und deren Verschiedenheit gekennzeichnet ist (siehe Abbildung 8).

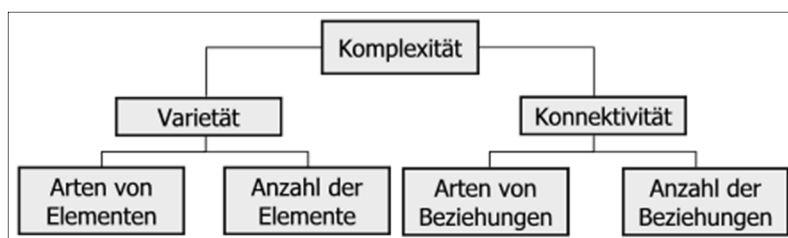


Abbildung 8: Komplexität nach (Steinmeier, 1999)

Durch immer höhere Anforderungen an die Effizienz und Wirtschaftlichkeit werden zwar die technischen Erzeugnisse komplexer, viel stärker ist aber der Zuwachs an Komplexität bei deren Entwicklungsprozessen. Ein gutes Beispiel dafür ist ein modernes Automatikgetriebe in einem Fahrzeug. Ein Automatikgetriebe muss nach wie vor in erster Linie eine hohe Leistungsdichte und einen guten Wirkungsgrad aufweisen und zur nachhaltigen Reduzierung des Gesamtverbrauchs beitragen. Daneben rücken die Anforderungen an die Schaltzeit und den Schaltkomfort in den Vordergrund. Nicht weniger aufwendig ist die Optimierung der akustischen Merkmale eines Getriebes, denn sie erfordert die Berücksichtigung nahezu aller Getriebeelemente und die Anpassung der Mikrogeometrie mit Hilfe einer detaillierten Lastverteilungsberechnung der Verzahnung. Die Vielfalt oft gleichgewichtig nebeneinanderstehender Anforderungen führt damit ebenfalls zur Steigerung der Prozesskomplexität. Aber nicht nur die Vielfalt der Elemente eines Erzeugnisses und enge Beziehungen untereinander erfordern im Hinblick auf eine produktivere Wissensverarbeitung neue Ansätze, sondern auch die Fülle des nicht zu vernachlässigenden externen und interdisziplinären Wissens, das die Vernetzung der Prozesse noch weiter fördert.

### 2.1.3 Informationsmanagement und Dokumentation

In der Abbildung 9 ist der Konstruktionsprozess dargestellt, so wie er heute in den Unternehmen vorzufinden ist. Zahlreiche unmittelbar Beteiligte sowie interne und externe Einflüsse und Auswirkungen machen die Arbeitsumgebung eines Ingenieurs zu einem Umschlagplatz des Wissens. Europäische Organisationen, Regierungen, die Normung und verschiedene wissenschaftliche Vereine produzieren darüber hinaus stetig neue Vorschriften und Regeln. Diese ständig wachsende Regelungs- und Informationsflut führt dazu, dass die Konstrukteure immer mehr Zeit mit dem Wissensmanagement und der Dokumentation von Ergebnissen verbringen. Schon Ende der 70er Jahre wurde ein prozentueller Anteil von bis zu 20 % der Arbeitszeit eines Ingenieurs für Informationsbeschaffung diagnostiziert (Hubka, 1976).

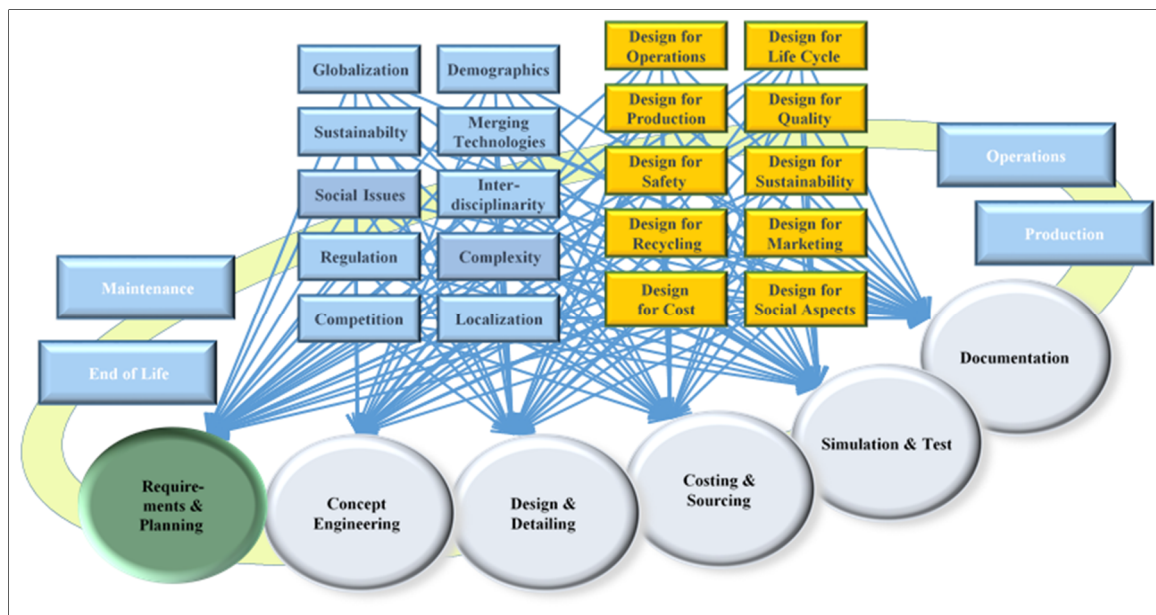


Abbildung 9: Konstruktionsprozess von heute (Gründer, 2010)

Ein großer Teil des Aufwands für das Management von Informationen und Dokumenten wird dabei von der Gesetzgebung und Normung verursacht. Zertifikate und Nachweise für die Nachhaltigkeit, Sicherheit, Zuverlässigkeit und Qualität erfordern einen zunehmenden Aufwand und viel Papierarbeit. Dennoch kann auf diese Arbeit nicht verzichtet werden. Sie muss vielmehr nahtlos in die Arbeitsprozesse eingefügt werden. Bezogen auf die am Anfang des Kapitels gestellten Fragen können folgende Aussagen getätigt werden:

Die Produktplanung ist durch die steigende Produktkomplexität und die steigende Anzahl der zu erfüllenden Anforderungen und Kundenwünsche selbst zu einem sehr komplexen Prozess geworden. Die wachsende Anzahl hybrider Produkte, innovativer Fertigungsprozesse und neuer Werkstoffe führt zu neuen interdisziplinären und hochintegrierten Entwicklungsmethoden. Die Folgen der Vernetzung hochbelasteter Maschinenelemente und Baugruppen sind so vielfältig, dass auch kleine Modifikationen einzelner Komponenten die wiederholte Verifikation des kompletten Systems zur Folge haben können. Dabei gewinnen Dokumentationsprozesse als Grundlage für einen kontinuierlichen Verbesserungsprozess weiter an Bedeutung.

Erst im Rahmen einer durchgängigen informationstechnischen Vernetzung entlang des kompletten Entwicklungsprozesses können viele Optimierungspotentiale hinsichtlich Qualität, Kosten und Innovation identifiziert werden. Somit kommt dem Management aller für die Entwicklung relevanten internen und externen Informationen, und damit deren gezielte und auf jede Entwicklungsaktivität abgestimmte Bereitstellung eine zentrale Aufgabe im Produktentstehungsprozess zu (Vorest AG, 2015).



### 2.1.4 Wissensmanagement

Wissen entsteht, wenn Daten, Dokumente, eigene Erfahrungen und Ergebnisse empirischer Untersuchungen kognitiv von Menschen verarbeitet werden. Das erfolgt durch die Herstellung der Relationen zwischen vielen Informationen, deren Formalisierung sowie der Abspeicherung der Erkenntnisse in einer für die Wissensverarbeitung geeigneten Form (Müller, 2009). Zu den wichtigsten Wissensquellen in einem Unternehmen zählen die Experten, die eine jahrelange Erfahrung mit sich tragen. Je größer ein Unternehmen ist, desto wichtiger ist dabei das Thema Wissensmanagement, denn in kleinen Unternehmen kann der Erfahrungsaustausch immer noch über den engen persönlichen Kontakt stattfinden, was bei großen Firmen allein aus organisatorischen Gründen schwer möglich ist (Herbst, 2000). Demzufolge lässt sich häufig beobachten, dass in Unternehmen unklar ist, wo die Experten sitzen, wer über welches Wissen verfügt und an welchen Projekten sie zurzeit arbeiten oder in der Vergangenheit tätig waren (Jung, 2001).

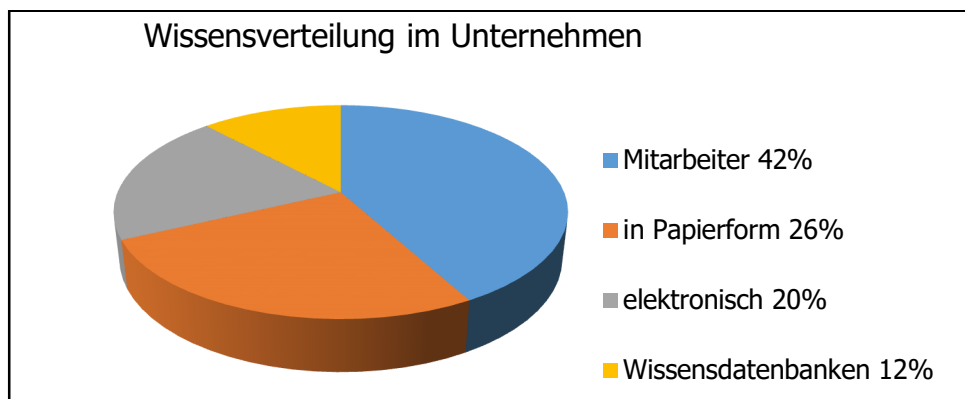


Abbildung 10: Wissensverteilung im Unternehmen (Jung, 2001)

Aus Abbildung 10 geht hervor, wie wichtig die Wissenskomponente ‚Mitarbeiter‘ ist. Denn fast die Hälfte des im Unternehmen befindlichen Wissens ist für die anderen Mitarbeiter dieses Unternehmens nicht zugänglich (Jung, 2001). Als Folge werden viele erfolgreiche Projekte, Lösungen und Entscheidungen nicht wiederverwendet, und es geht durch die Wiederbeschaffung eigentlich schon vorhandenen Wissens viel Zeit verloren. Abhilfe kann hier ein Wissensmanagement schaffen, das vorhandene Erkenntnisse und Erfahrungen auch der prozessualen Ebene verfügbar macht, denn durch den Aufbau und die Anwendung eines Wissens-Repository kann die Effizienz der künftigen Entwicklungsprojekte durch die Wiederverwendung der einzelnen Teillösungen nachhaltig gesteigert werden (Abbildung 11).

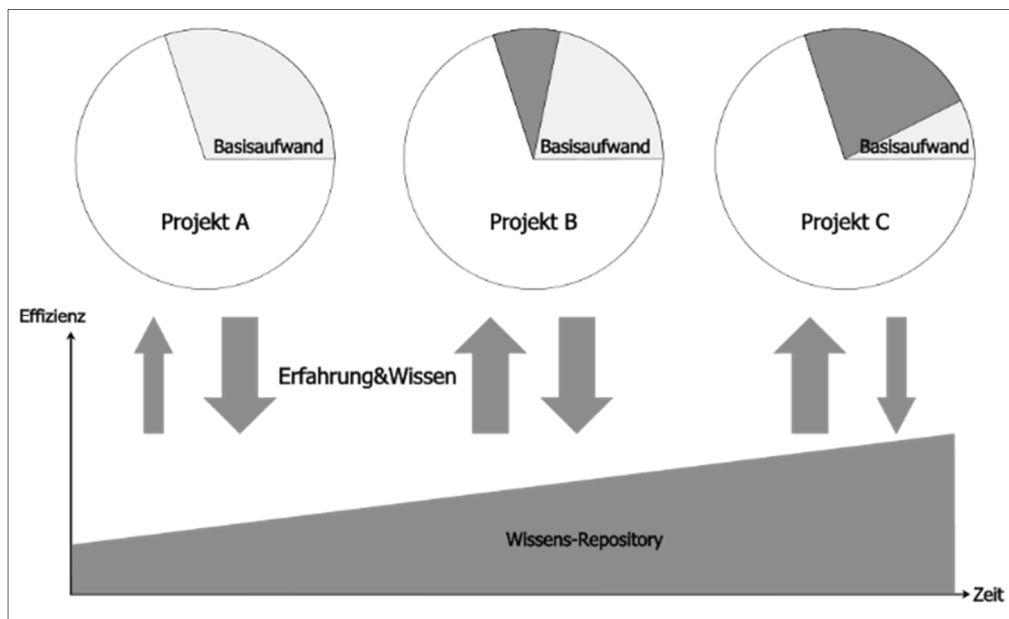


Abbildung 11: Effizienzsteigerung durch ein Wissens-Repository (Kiffe und Bock, 2013)

### 2.1.5 Analytische Berechnung in der Konstruktion

Berechnungen sind zum einen sehr wichtig in der Konzeptphase, denn diese helfen die Gestaltelemente gezielt festzulegen und damit auch die Entwicklungszeiten zu verkürzen und zum anderen als Absicherungswerkzeug in der Detaillierungsphase. Besonders notwendig sind die Berechnungen, wenn bestimmte Produkteigenschaften nicht visuell z.B. mit Hilfe eines CAD-Modells verifiziert werden können. Dann sind Algorithmen erforderlich, um die Produkteigenschaften in Bezug auf die gestellten Anforderungen oder Randbedingungen festzulegen, indem die Beziehungen zwischen Gestalt- und Werkstoffeigenschaften sowie den entsprechenden Anforderungen als Berechnungsmodell formuliert und anschließend mathematisch formalisiert werden.

Abhängig von den Zielen werden die Berechnungen in funktionsorientierte und gestaltorientierte Berechnungen unterteilt. Die funktionsorientierten Berechnungen dominieren in der Konzeptphase und helfen zum Beispiel die optimalen Wirkprinzipien für die Teilfunktionen zu ermitteln. Dazu zählen u.a. die Kinematik-Berechnungen von Getrieben und die damit verbundene Bestimmung von Radsatzvarianten und deren Anordnung (VDI 2223:2004).

Die Entwurfsphase sowie die Ausarbeitung eines Produktes oder eines Bauteils sind gleichermaßen von den gestaltorientierten Berechnungen geprägt. Hier geht es vor allem darum, die anforderungsgerechten Festlegungen bezüglich der Gestalt- und Werkstoffeigenschaften zu definieren. Dazu gehören beispielsweise Aufgaben der Dimensionierung von Bauteilen unter äußeren Einflüssen oder die Kostenberechnungen der Herstellung unter bestimmten fertigungstechnischen Randbedingungen (VDI 2223:2004).

Im Hinblick auf die Wirkungsweise sowie auf das Gestalten unterscheiden sich die Berechnungen in Vorausberechnung (Auslegung) und Nachrechnung.

Bei der Auslegung werden ausgehend von den gestellten Randbedingungen (Anforderungen bzw. bekannte Produkteigenschaften) die gestaltgebenden sowie die stofflichen Eigenschaften berechnet bzw. ermittelt. Als Beispiel lässt sich folgende Berechnung zur Ermittlung des minimal erforderlichen Durchmessers für eine torsionsbeanspruchte Welle nach DIN 743-1:2012-12 (z. B. eine Kardanwelle) aufzeigen:

$$d_{\min} = \sqrt[3]{\frac{M_{t\max} \cdot 16}{T_{t\max} \cdot \pi}} \quad (2-1)$$

Mit dem überschlägig ermittelten Wellendurchmesser können anschließend weitere Komponenten wie Radnaben und Lager vordimensioniert werden.

Bei der Nachrechnung ist von der bereits festgelegten Geometrie bzw. den Werkstoffeigenschaften eines Bauteils auszugehen, um dann deren Einfluss auf die Funktion und die Kosten zu ermitteln. Am häufigsten werden die Nachrechnungen zum Nachweis der minimal erforderlichen Lebensdauer oder der Festigkeit eines Bauteils durchgeführt (VDI 2223:2004), z.B. der Nachweis des Vermeidens von Dauerbrüchen an einem kritischen Wellenquerschnitt unter Berücksichtigung von Zug/Druck, Biegung, Torsion und unter der Annahme der Phasengleichheit nach DIN 743-1:2012-12:

$$S = \frac{1}{\sqrt{\left(\frac{\sigma_{zda}}{\sigma_{zdADK}} + \frac{\sigma_{ba}}{\sigma_{baDK}}\right)^2 + \left(\frac{T_{ta}}{T_{taDK}}\right)^2}} \quad (2-2)$$

Außerdem kann die Nachrechnung auch erforderlich sein, wenn im Laufe des Konstruktionsprozesses einige Bauteileigenschaften aus Kosten- oder Konstruktionsgründen so stark geändert werden, dass die zuvor festgestellte Anforderungserfüllung in Frage gestellt wird und neu überprüft werden muss (VDI 2223:2004).

### 2.1.6 IT-Unterstützung in der Produktentwicklung

Im Laufe der Produktentwicklung werden Dutzende von Auslegungen, Berechnungen und Optimierungen durchgeführt. Die dabei entstehenden Strukturen und Geometrien der mechanischen und mechatronischen Erzeugnisse werden dann mit ihren Dimensionen, Kennzahlen und Annotationen als dreidimensionale CAD-Modelle abgespeichert. Daraus werden schließlich Listen und Fertigungsdokumente für die Planung, die Herstellung und die Montage abgeleitet. Die Daten müssen den Planungsprozess dokumentieren, darüber hinaus aber auch für jede weitere Produktlebensphase

verfügbar sein. Ziel ist es, einen maximalen Nutzen und größtmögliche Effizienz unter wechselnden Einsatzbedingungen und fortschreitender Abnutzung zu ermöglichen (Eigner und Stelzer, 2009).

Als Wissens- und Datenbasis während der Entstehung, aber auch als Integrationsdrehscheibe für das lebenslange Monitoring eines Produktes dienen gegenwärtig PDM/PLM-Systeme. Mit solchen Softwaresystemen haben die Konstruktionsabteilungen einen direkten Zugriff auf fast alle produktbezogenen Daten und Dokumente ihres Unternehmens. Vor allem konzentrieren sich die heutigen PDM/PLM Lösungen auf die Verwaltung von Informationen aus der virtuellen Produktentstehung und bedienen sich dabei der Anwendungen CAD, CAE und CAM (Abramovici und Schulte, 2004).

Die CAD-Systeme fanden bis heute ihre Anwendung hauptsächlich auf der geometrischen und funktionalen Ebene. Damit werden die Lösungen der Aufgaben in der Entwurfs- und Ausarbeitungsphase unterstützt. In den letzten Jahren stieg aber auch die Anzahl der Beispiele, bei denen Entwurf, Berechnung sowie wissensbasierte Systeme ins CAD integriert wurden. Dies ist nicht zuletzt auch der raschen Entwicklung der virtuellen Produktentwicklung zu verdanken, denn diese fordert die nahtlose und möglichst frühzeitige Integration der Berechnung und Simulation in den Entwicklungsprozess. Eine wichtige Voraussetzung für das erfolgreiche Zusammenwirken von verschiedenen Systemen ist das Offenhalten und die Standardisierung der Schnittstellen (Bullinger, et al., 2009).

Neben den rechnergestützten Standardwerkzeugen existieren in der technischen Produktentwicklung oft zahlreiche weitere konstruktionsunterstützende Verfahren. Denn viele Problemstellungen der Auslegung und Gestaltung von Maschinen erfordern spezialisierte Berechnungs- und Simulationsprogramme. Als Folge hiervon müssen immer mehr Aktivitäten vernetzt und automatisiert werden, um den aufwendiger werdenden Informationsfluss zeitlich und datentechnisch beherrschen zu können. Auf diese Weise und mit Hilfe neuer prozessorientierter Benutzeroberflächen zur Kopplung von Simulationsanwendungen gelingt es, Konstruktionsabläufe über mehrere Softwareapplikationen hinweg zu schaffen (siehe Abbildung 12) (Gründer und Polyakov, 2011).

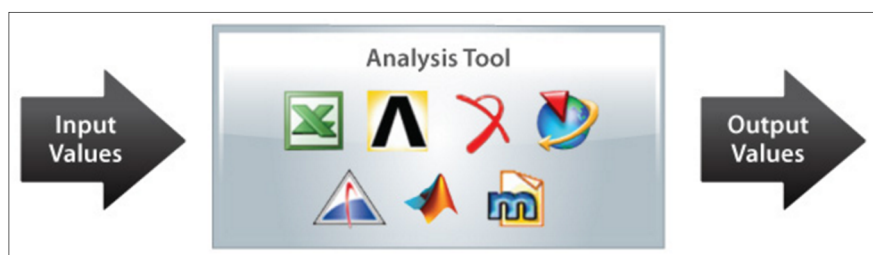


Abbildung 12: Prinzip ModelCenter® Integrate, Phoenix Integration, Inc (Phoenix integration, 2016)

Die prozessorientierte Kopplung verschiedener Programme spart zwar kostbare Entwicklungszeit und minimiert das durch Medienbrüche und manuelle Datenübertragung verursachte Fehlerpotential, sie versäumt es aber, eine aktive Wissensverarbeitung in die Abläufe zu integrieren. Während die Produktion bereits in den Phasen der Mechanisierung und Automatisierung weitgehend fließend organisiert wurde und aufgrund der dabei gegebenen Transparenz und Modularität neue Technologien jederzeit integrieren kann, hat die Produktentwicklung das volle Potenzial von Workflows bei weitem noch nicht erkannt und ausgenutzt. Wo dennoch Simulationsprozesse vernetzt werden und unmittelbar miteinander kommunizieren, beruhen diese Abläufe zumeist auf Entwicklungsumgebungen, die ihre Ergebnisse nur in kodierter verschlüsselter Form zur Verfügung stellen (siehe Abbildung 13) (Polyakov und Gründer, 2016, Polyakov und Gründer, 2015).

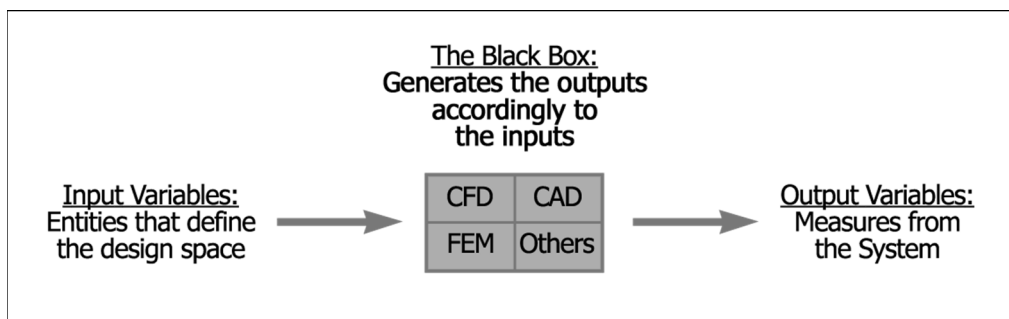


Abbildung 13: Prinzip von dem Tool modeFRONTIER, adaptiert nach (ESTECO SpA, o.D.)

## 2.2 Konstruktionsmethoden in Maschinenbau

In diesem Kapitel sollen die gegenwärtig etablierten Ansätze des methodischen Konstruierens kurz erläutert und analysiert werden. Dazu zählen die Methoden nach Koller (Koller, 1994, Pahl und Beitz, 1997, Ehrlenspiel, 1995, VDI 2221:1993).

### 2.2.1 Vorgehensmodelle

Ausgangspunkt jeder Produktentwicklung ist eine auf das Entwicklungsziel gerichtete Liste mit Anforderungen. Die hier geforderten Produktmerkmale können im Verlauf der Planungsprozesse insbesondere bei agilen Vorgehensweisen zahlreiche Anpassungen erfahren. Haupttätigkeiten bei der Planung und Konstruktion sind das Konzipieren, Entwerfen und Ausarbeiten der Produkte (Pahl und Beitz, 1997). Ein Schwerpunkt liegt dabei auf der Beachtung von Normen und Gestaltungsrichtlinien, wonach die Konstruktion festigkeitsgerecht, montagegerecht und werkstoffgerecht sein muss (Weishaupt, 1998, Koller, 1994).

Eine erweiterte Sicht (Ehrlenspiel, 1995) stellt die Produkterstellung als einen gemeinsamen Prozess dar, welcher die Phasen von der Ideensuche, bei der das Produkt zunächst definiert wird, über den Auftrag bis hin zur Auslieferung des Produkts an den Nutzer einschließt. Das Entwickeln und Konstruieren als Kern der Produkterstellung nimmt dabei in Zusammenarbeit mit der Marktanalyse,

dem Vertrieb, der Produktionsplanung, den Zulieferern, dem Controlling und dem Qualitätsmanagement die führende Stellung ein. Mit der integrierten Produkterstellung wurde die Grundlage für einen durchgängigen, alle Disziplinen berücksichtigenden Informationsfluss im Produktentstehungsprozess geschaffen. Sie gilt deshalb als Basis für Produkt-Daten-Management (PDM) - Systeme, wie sie im Umfeld der CAX-Technologien zum Standard geworden sind. Die integrierte Produktentwicklung stellt die Entwicklungsprozesse, das Projektmanagement, die Werkzeuge, den Vorgehenszyklus, das Kostenmanagement und den Menschen in einen strategischen, methodischen und organisatorischen Zusammenhang (Frei, 2002, Leemhuis, 2005, Diehl, 2009).

Aufgrund der heute immer kürzer werdenden Durchlaufzeiten wird das Hauptaugenmerk der Rationalisierungsanstrengungen neben der fachlichen und interdisziplinären Konsistenz neuer Produkte zunehmend auf die Domäne übergreifender informationstechnischer Datendurchgängigkeit gelegt (Weishaupt, 1998). Die Ausweitung der Entwicklungsprozesse auf den gesamten Produktlebenszyklus erfordert die Berücksichtigung einer Fülle von Informationen, Erfahrungen und Erkenntnissen und damit die Beteiligung von immer mehr Stakeholdern. Andererseits sichert diese frühe Phase im Produktlebenszyklus dem Unternehmen die Grundlage für die Erreichung seiner Kosten-, Zeit- und Qualitätsziele (Westkämper, 2006).

Diese genannten Methoden sehen den Konstruktionsprozess als eine Abfolge bestimmter Tätigkeiten, welche zu wenigen wichtigen Phasen mit zugehörigen Meilensteinen zusammengefasst werden können. Dabei ergeben sich die Eingangsdaten einer Phase aus den Ergebnissen der vorherigen Phase, wodurch deren Reihenfolge vordefiniert ist. Sie können sich aber in einem iterativen Prozess wiederholen, wobei die Ergebnisse einer Phase durch die Ergebnisse einer nachkommenden Phase präzisiert und deshalb möglicherweise rückgekoppelt werden müssen.

Einen solchen insbesondere für die Phase der Auslegung bestimmten Prozess betrachtet die Dissertation von Martin Frenzel an der TU Dresden (Frenzel, 1980), die sich mit dem Aufbau von problemorientierten selbstorganisierenden Systemen in der Elektrotechnik beschäftigte. Seine auf „Arbeitsmodulen“ und „Steuermodulen“ aufgebaute Programm- und Datenstruktur versprach die nötige Flexibilität und Wiederverwendung von Teilprozessen, wobei er sich bereits für eine einheitliche Beschreibung von Problem- und Steuerdaten durch Deskriptoren einsetzte. Voraussetzung waren damals wie heute detaillierte Kenntnisse über die zugehörigen technischen Lösungsprozesse. Zusätzlich mussten die damals zur Bewältigung umfangreicher Berechnungen notwendigen rechentechnisch organisatorischen Maßnahmen einbezogen werden. Obwohl diese Arbeit wichtige Hinweise für eine prozessorientierte Betrachtung von Konstruktionsberechnungen lieferte, setzte die Informationstechnik den vorgeschlagenen Methoden noch enge Grenzen. Erst mit den fortschreitenden Datenbanktechnologien und grafischen Oberflächen wurden diese Einschränkungen weniger bedeutsam.

Die grundlegenden Meilensteine bei diesen phasenorientierten Methoden werden in folgende Tätigkeitsfelder unterschieden (vgl. Conrad, 2013):

- Planung (Aufgabenstellung klären, informative Festlegungen)
- Konzept (Konzept entwickeln, prinzipielle Festlegungen)
- Entwurf (Entwurfsarbeit durchführen, gestalterische Festlegungen)
- Ausarbeitung (Unterlagen ausarbeiten, herstellungstechnische Festlegungen)

Am Beispiel einer Getriebeentwicklung sollen die genannten Schritte im Folgenden beispielhaft erklärt werden.

Als erstes wird unterschieden, ob die prinzipielle Lösung in der Aufgabenstellung vorgegeben ist. Ist das der Fall, müssen die gestellten Anforderungen wie zum Beispiel die Angaben zu Drehzahlen, Leistungen, Wirkungsgraden, usw. auf Vollständigkeit geprüft werden. Anderenfalls müssen unter den Gesichtspunkten Funktion, Wirkungsweise, Geometrie, Werkstoff usw. Lösungsprinzipien auf der Basis der vorhandenen Anforderungen erarbeitet werden. Dabei werden die Aufgaben in Teilaufgaben zerlegt. Bei einem Getriebe werden zum Beispiel abhängig unter anderem von der Antriebsart und der gewünschten Gesamtübersetzung die Konzepte für einzelne Radsätze, deren Anzahl und Anordnung vorgeschlagen oder erarbeitet. Anschließend kann mit dem Entwurf begonnen werden.

Bei dem Entwurf handelt es sich um die Festlegung der konstruktiven Ausführungen der einzelnen Getriebekomponenten (Radsätze, Wellen, Lager, Gehäuse usw.) und anschließend des kompletten Getriebes. Dies erfolgt in einem iterativen Vorgehen, bei dem auch die Gliederung der prinzipiellen Lösungen in realisierbare Module stattfindet. Dieser Schritt ist notwendig, um nach dem groben Systementwurf die Konkretisierung der Baugruppen voneinander getrennt und möglichst parallel durchführen zu können. Die Phasen der Anforderungsdefinition und des Entwurfes sind deshalb besonders kritisch, weil die meisten Fehler in diesen Phasen unterlaufen und umso teurer sind, je später sie entdeckt und behoben werden (Institut für Technologie und Arbeit (ITA), 2013, Endres und Rombach, 2003).

Nach dem groben Systementwurf werden die einzelnen Komponenten und Baugruppen mit den nötigen Abmessungs- sowie Materialdaten in einem Detailentwurf versehen.

Als Letztes werden die fertigungstechnischen sowie montagetechnischen Unterlagen angefertigt, wobei eine wiederholte und präzise Überprüfung der Getriebeparameter erfolgt. Als Ergebnis liegen alle im Rahmen des Konstruktionsprozesses geschaffenen Unterlagen für die Auslegung, Produktion und Nutzung des Produktes in Form einer Produktdokumentation vor (Ehrlenspiel, 2009).

Das Vorgenannte bildet die Grundlage für eine „Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte“, die in der Richtlinie VDI 2221:1993 (siehe Abbildung 14) beschrieben wird.

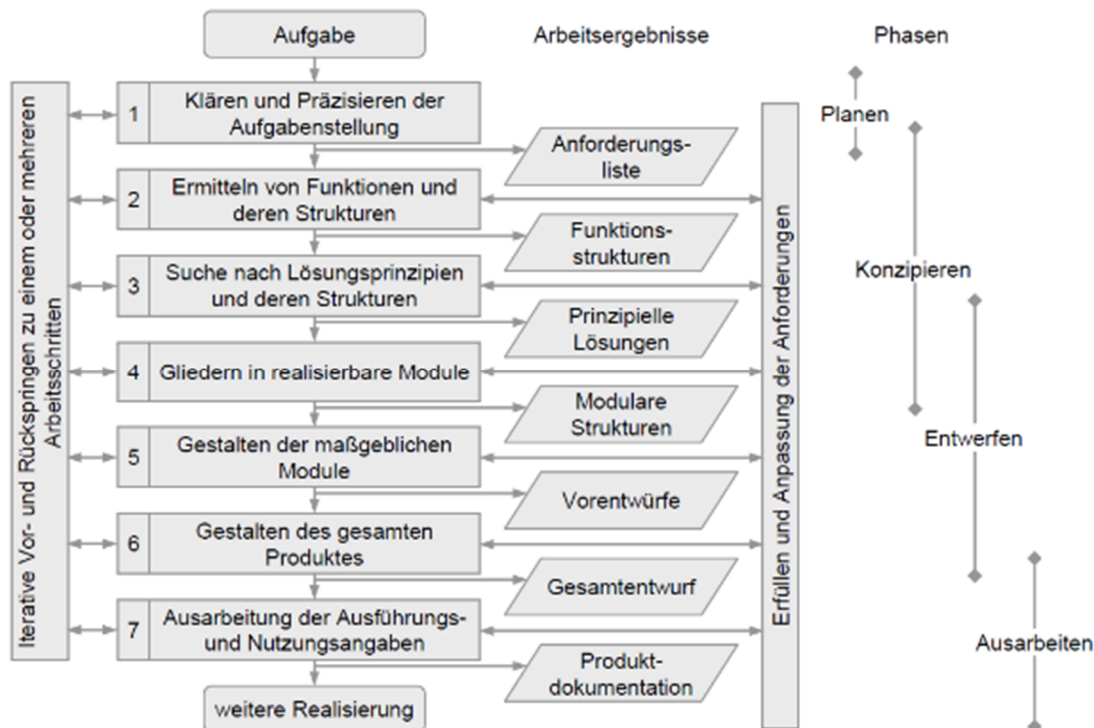


Abbildung 14: Vorgehensweise beim Konstruieren nach VDI 2221:1993

Die Methodik und Sichtweisen treffen jedoch nicht mehr auf die Gesamtheit der heutigen maschinenbaulichen Erzeugnisse zu, da diese oft sehr komplexe Systeme darstellen, bei denen die Mechanik durch andere Disziplinen wie Informatik und Elektrotechnik überlagert wird (Möhringer, 2003). Die Entwicklung heutiger Produkte erfordert in vielen Fällen eine deutlich breitere disziplinübergreifende Sicht.

2004 wurde eine VDI-Richtlinie (VDI 2206:2004) veröffentlicht, die durch den informationstechnischen Ansatz des Systems Engineering getragen wird. Diese stellt eine Entwicklungsmethodik für mechatronische Systeme dar. Das Ziel dieser normierten Methodik war, einen domänenübergreifenden Systementwurf zu verwirklichen, der bereits in den früheren Phasen der Produktentwicklung ansetzt (Möhringer, 2003).

Für die vorliegende Arbeit sind von den maßgebenden Kennzeichen des Vorgehensmodells für die Entwicklung von mechatronischen Systemen nach VDI-2206 insbesondere „Das V-Modell als Makrozyklus“ und Prozessbausteine für wiederkehrende Arbeitsschritte von Bedeutung. Diese werden in den nachfolgenden Kapiteln näher beschrieben.



### 2.2.2 Das V-Modell der VDI-2206

Das generische Vorgehen zur Entwicklung mechatronischer Systeme basiert in VDI 2206 auf der Weiterentwicklung des aus der Softwareentwicklung stammenden V-Modells, welches hier die Reihenfolge der grundlegenden Schritte bei der Entwicklung mechatronischer Systeme beschreibt. Ähnlich wie in der VDI-2221:1993 steht am Anfang der Prozesskette ein bestimmter Entwicklungsauftrag mit entsprechenden Anforderungen (Ponn und Lindemann, 2011). Gefolgt vom Systementwurf, dem domänenspezifischen Entwurf und der Systemintegration (Zingel, 2013) bis zum fertigen Produkt sind die einzelnen Phasen in „V-Form“ graphisch angeordnet (siehe Abbildung 15). Die kontinuierliche Eigenschaftsabsicherung (Verifikation und Validierung) im rechten Zweig wird durch die V-förmige Anordnung gestärkt. Die einzelnen Integrationsschritte von verschiedenen Systemkomponenten und dem System im Ganzen müssen den gegenüberliegenden Entwürfen im linken Zweig entsprechen (Diehl, 2009). Dabei ist sicherzustellen, dass die tatsächlichen Systemeigenschaften den gewünschten bzw. geforderten Systemeigenschaften genügen.

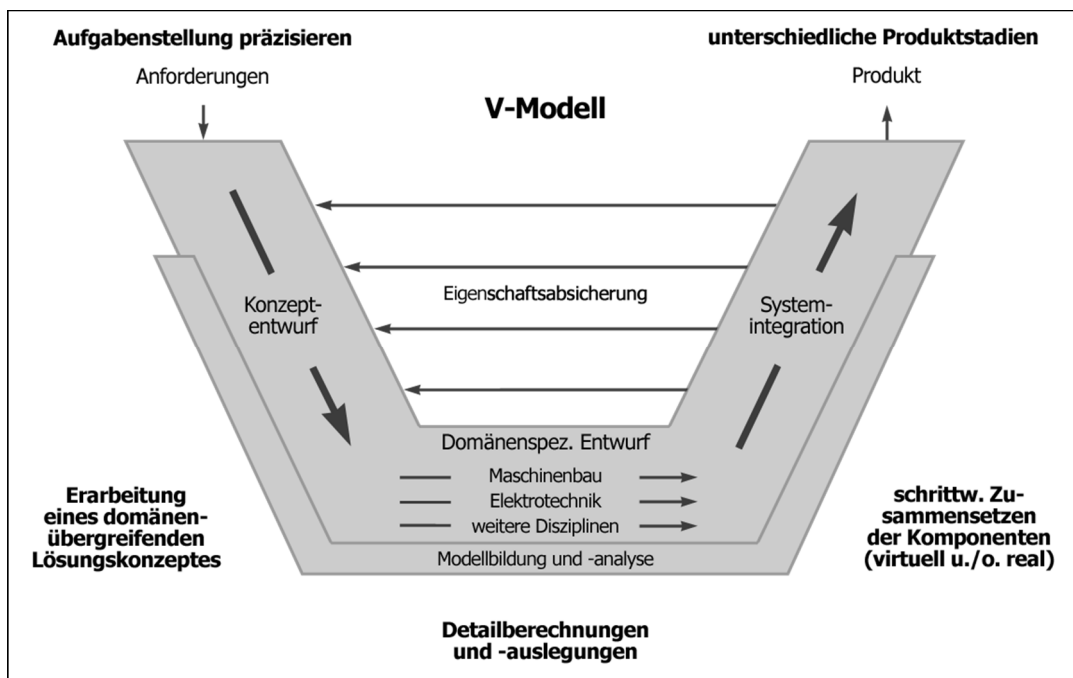


Abbildung 15: V-Modell für den Entwurf mechatronischer Systeme (Panreck, 2014)

Das Systemkonzept, welches die wichtigsten physikalischen und logischen Wirkungsweisen des zu entwickelten Produktes beschreibt, ist im linken Zweig des Diagramms angesiedelt. Auf seiner Basis werden anschließend die disziplinspezifischen Konkretisierungen (Teilentwürfe), meistens noch getrennt nach Domänen (Maschinenbau, Elektrotechnik, Informationstechnik, etc.) vorgenommen (Ponn und Lindemann, 2011). Zum Einsatz kommen dabei die jeweiligen fachrelevanten Entwurfsmethoden und Werkzeuge (Gausemeier, et al., 2001). Danach folgt die Integration der domänenspezifischen Teillösungen zu einem Gesamtsystem, wobei die Eigenschaften und das Verhalten des Gesamtsystems analysiert und mit den aufgestellten Anforderungen verglichen werden

(Baumberger, 2007). Der Hauptvorteil des V-förmig angelegten Modellverlaufs wird besonders bei der Systemintegration in der Absicherung der theoretisch gelegten Ansätze durch die praktisch realisierten Lösungen deutlich. Denn erst hier werden die Verknüpfungen zwischen den entwerfenden Aufgaben im linken Zweig und den konstruktiven Implementierungsaufgaben im rechten Zweig konsequent verfolgt (Störrle, 2005).

Die Komplexität der mechatronischen Produkte fordert meistens einige wiederholte Durchläufe des hier beschriebenen Makrozyklus, was auch die ineinander verschachtelten „V-Verläufe“ kennzeichnen sollen (Ponn und Lindemann, 2011).

Trotz der kontinuierlichen Eigenschaftsabsicherung weist das V-Modell besonders im Bereich des disziplinspezifischen Entwurfes einen gewissen sequentiellen Charakter auf (Zingel, 2013). Zum Beispiel ist es innerhalb eines Makrozyklus nicht vorgesehen, dass die Ergebnisse des domänenspezifischen Entwurfes samt der dort gewonnenen Erkenntnisse und eventueller Änderungsvorschläge zum Gesamtsystem in irgendeiner Form in den Systementwurf zurückfließen können (vgl. ebd.). Ähnlich ist es auch bei den Anforderungen, welche ganz am Anfang des Entwicklungsprojektes festgelegt wurden. Sie können erst nach der komplett durchgeführten Systemintegration korrigiert werden.

### 2.2.3 Modularisierung (Prozessbausteine)

Einen weiteren wichtigen Lösungsansatz für die Systemintegration bietet die VDI 2206 mit der Definition des Prozessbausteines: „Eine abgeschlossene Einheit von Tätigkeiten, die dazu dienen, ein bestimmtes Zwischenziel zu erreichen. Ein Prozessbaustein enthält darüber hinaus eine Beschreibung der Eingangs- und Ausgangsinformationen, Klassifizierungskriterien und Zusatzinformationen (wie unterstützende Methoden oder erforderliche Kompetenzen), die das Arbeiten mit dem Prozessbaustein unterstützen.“ (VDI 2206:2004)

Indem regelmäßig auftretende Aktivitäten identifiziert und in Form von vordefinierten Teilprozessen beschrieben werden, wird die Grundlage zur Modularisierung des Entwurfsprozesses geschaffen. Die Wiederverwendung von den Teilen des Entwurfsprozesses verstärkt das Wissensmanagement im Unternehmen und trägt zur Nachhaltigkeit des Prozesses insgesamt bei.

Als Beispiele bietet die Richtlinie entsprechende Prozessbausteine wie etwa den Systementwurf, die Modellbildung und -analyse, den domänenspezifischen Entwurf, die Systemintegration und die Eigenschaftsabsicherung, welche den Phasen bzw. Meilensteinen und Haupthandlungstätigkeiten des Makrozyklus entsprechen. Eigentlich sollen die Prozessbausteine deutlich präziser und kleiner sein und bis hin zur funktionalen Ebene reichen (zum Beispiel: Antreiben, Steuern/Regeln, Messen, Übertragen usw.). Zusammen mit der Darstellung der einzelnen Funktionen als Block-Schaubild könnte das die

Integration der Partialmodelle der einzelnen Disziplinen zu einem Gesamtsystem deutlich flexibilisieren.

Allein für die erfolgreiche Integration von Partialmodellen fehlt eine gemeinsame Sprache, mit der das System als Ganzes samt Verhalten abgebildet werden könnte. Dieses Fehlen einer gemeinsamen Sprache für das Systemverhalten macht den koordinierten Vergleich zwischen gestellten und erfüllten Anforderungen sowie den realisierten Teilfunktionen und den einzelnen disziplinspezifischen Baugruppen immer noch sehr zeit- und ressourcenaufwendig.

## 2.2.4 Prozessorientierte Betrachtung der Konstruktion

Im Bereich des Qualitätsmanagements wurde die Bedeutung eines durchgängigen Prozessmanagements längst erkannt. Die Zertifizierung nach ISO 9001, der Norm für Qualitätsmanagementsysteme, verspricht eine Erhöhung der Gewinne, eine Steigerung der Leistung sowie den Ausbau strategischer Erfolgsfaktoren in einem Unternehmen (DIN EN ISO 9001:2015-11). Die wesentlichen möglichen Vorteile des prozessorientierten Ansatzes werden darin wie folgt gekennzeichnet:

- größere Möglichkeit der Konzentration auf die Schlüsselprozesse und Verbesserungspotentiale,
- bessere Planung durch einen systemangepassten Prozess,
- Leistungssteigerung durch effizientere Ressourcenverwendung und Reduzierung von funktionsübergreifenden Barrieren sowie
- mehr Vertrauen in Bezug auf Beständigkeit, Wirksamkeit und Effizienz von tatsächlichen Arbeitsabläufen seitens der Beteiligten (DIN EN ISO 9001:2015-11).

Zusammengefasst können die gewünschten Ergebnisse deutlich effizienter erzielt werden, wenn die Tätigkeiten, die zur ihrer Erzeugung nötig sind, alle als ein System mit aufeinander abgestimmten Prozessen funktionieren, verstanden und gesteuert werden. Da ein Qualitätsmanagementsystem aus vielen zusammenhängenden Prozessen besteht, wird es erst mit dem Verständnis darüber, wie das System insgesamt arbeitet und wie die Ergebnisse mit diesem System erzielt werden, möglich, das System zu optimieren und seine Leistung noch weiter zu steigern (vgl. ebd.).

Konstruktionsabläufe sind hochkomplex und unterliegen den gleichen Bestrebungen nach Effizienz, Nachhaltigkeit und Transparenz wie die Prozesse des Qualitätsmanagements. „Der Konstruktionsprozess ist in erster Linie ein informationsverarbeitender Prozess“ (Hubka, 1976). Die Darstellung des Konstruktionsablaufes als informationsverarbeitender Prozess (siehe Abbildung 16) hilft, dies zu veranschaulichen und die Relationen zu finden. Die Fülle der zu erfüllenden

Anforderungen wird benötigt, um den Prozess anzustoßen und die Produktbeschreibung samt Fertigungsunterlagen entsprechend als Ergebnisse des Prozesses zu erhalten

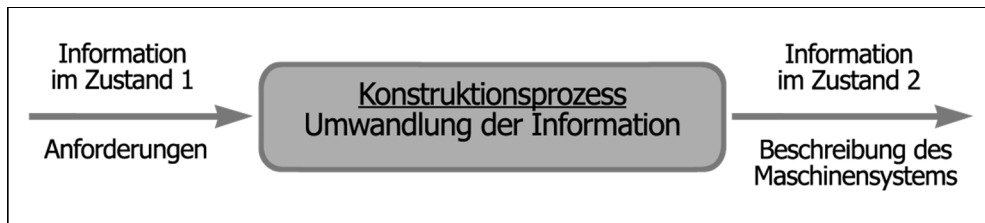


Abbildung 16: Vereinfachte Darstellung der Konstruktion als Informationsverarbeitungsprozess adaptiert nach (Hubka, 1976)

Dies entspricht auch der Definition eines Prozesses nach DIN EN ISO 9000:2015-11: als „Satz zusammenhängender oder sich gegenseitig beeinflussender Tätigkeiten, der Eingaben zum Erzielen eines vorgesehenen Ergebnisses verwendet“.

Damit ein Konstruktionsprozess als Abfolge einzelner Arbeitsschritte bzw. Teilprozesse abgebildet werden kann, und somit auch die Analyse ihrer gegenseitigen Einflussnahme möglich wird, muss zunächst die Identifikation der einzelnen Arbeitsschritte und deren Vernetzung untereinander vorgenommen werden. Diese Aufgabe der Zerlegung des Gesamtprozesses in sinnvolle Prozessbausteine ist von besonders großer Bedeutung, um später die Prozessmodelle flexibel steuern und gegebenenfalls optimieren zu können.

Ein Arbeitsschritt beim Konstruieren wird in (Feldhusen und Grote, 2013) durch das Versorgen eines Modells, welches eine bestimmte Eigenschaft des Produktes abbildet, mit den aufgabenspezifischen Daten beschrieben. Die anschließende Verarbeitung dieser Daten liefert die gesuchte Produkteigenschaft (siehe Abbildung 17).

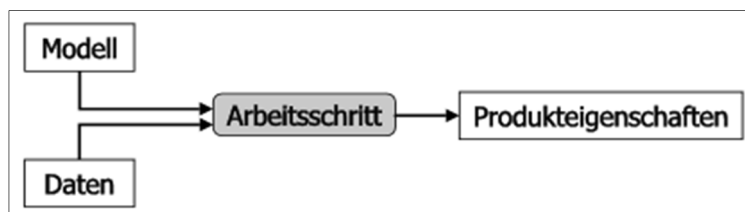


Abbildung 17: Schematische Darstellung eines Konstruktionsschrittes (ebd.)

Diese noch relativ abstrakte Formulierung kann am besten an einem Beispiel zur Bestimmung der Reibkraft veranschaulicht werden (siehe Abbildung 18).

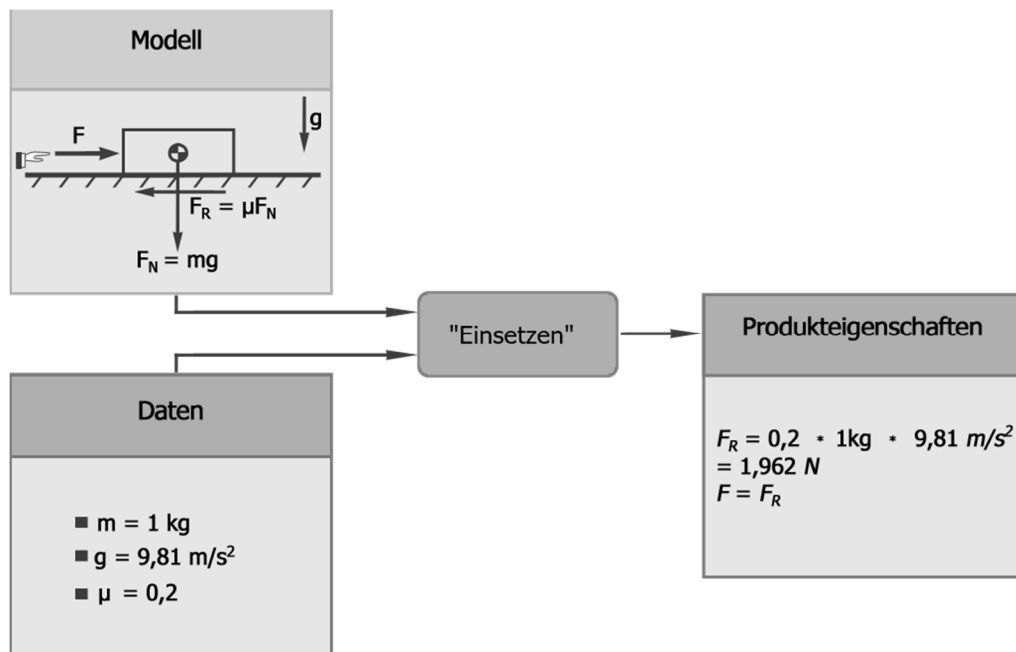


Abbildung 18: Konstruktionsarbeitsschritt am Beispiel "Reibkraft bestimmen" (Feldhusen und Grote, 2013)

Dabei werden die Daten wie Reibungskoeffizient  $\mu$  zusammen mit der Masse  $m$  und dem Erdbeschleunigungswert  $g$  nach dem Haftreibungsgesetz (siehe Gleichung (2-3)) zur Reibkraft  $F_R$  umgewandelt.

$$F_R = \mu \cdot F_N = \mu \cdot m \cdot g \quad (2-3)$$

In Bezug auf die Datenverarbeitung kann ein solcher Konstruktionsarbeitsschritt als das Ausrechnen einer Formel (Modell) mit entsprechenden Parametern (Daten) verstanden werden (Feldhusen und Grote, 2013) (siehe Abbildung 19).

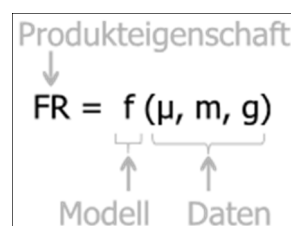


Abbildung 19: Bestimmung der Produkteigenschaft

Ein ähnliches Konzept verfolgt Zingel in seiner Dissertation (Zingel, 2013), mit der funktionszentrierten Modellbildung. Die technischen Funktionen spielen hier die zentrale Rolle bei der Entwicklung eines technischen Systems. In Abbildung 20 ist die von Zingel aufgestellte semantische Kopplung zwischen der heutigen Ingenieurssprache, die durch Ursachen, physische Komponenten und ihre Wirkung geprägt ist, und der funktionszentrierten Modelbildung dargestellt.

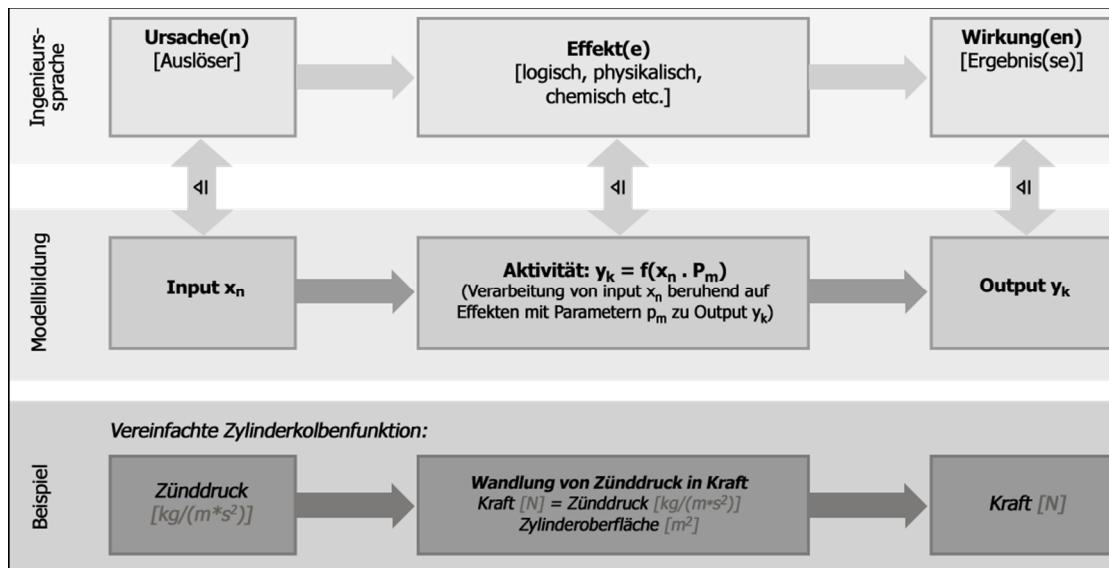


Abbildung 20: Semantische Kopplung des Funktionsverständnisses in der Ingenieursprache auf die funktionszentrierte Modellbildung (Zingel, 2013)

Die Kernaussagen aus dieser Abbildung definiert Zingel wie folgt:

- Das Anstoßen kann mit den Eingangsgrößen wie Stoff, Energie oder Information erfolgen, welche in Form von diskreten Größen oder kontinuierlichen Flüssen vorliegen.
- Die Aktivitäten übernehmen die Aufgabe, die Inputs zu Outputs zu verarbeiten.
- Die Ergebnisse (Outputs) einer Aktivität entsprechen der messbaren Wirkung einer Funktion, und können wiederum Inputs weiterer Funktionen sein (Zingel, 2013).

Da letztendlich alle Produkteigenschaften ermittelt werden müssen, wird das Produkt durch die Summe aller Konstruktionsschritte von der Idee bis zur Anfertigung der Produktunterlagen modellhaft abgebildet (siehe Abbildung 21).

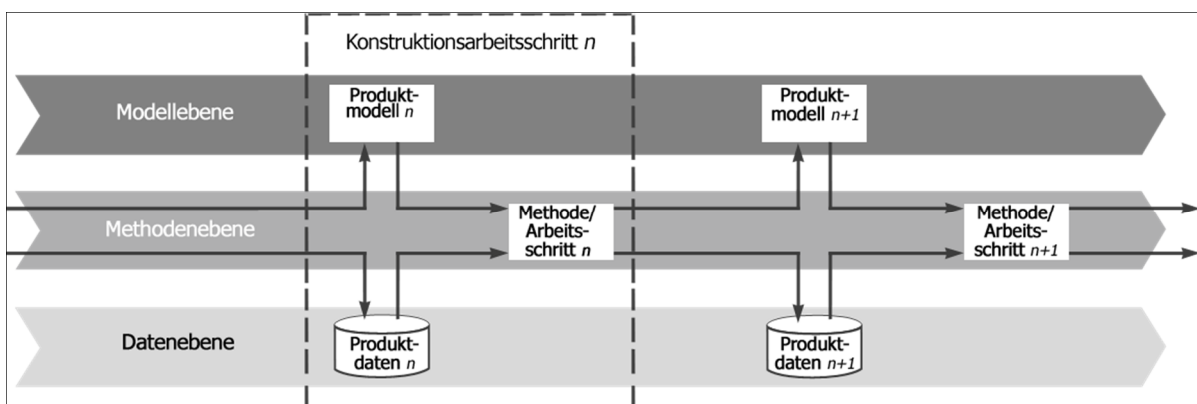


Abbildung 21: Auszug aus der Prozessorientierten Darstellung der Konstruktion (Feldhusen und Grote, 2013)

Die Existenz einer Möglichkeit, Konstruktionstätigkeiten aus prozessorientierter Sicht durchgängig zu beschreiben, würde zusätzlich zur obligatorischen Dokumentation der am Ende des Konstruktionsprozesses ermittelten Produkteigenschaften noch die Historie ihrer Entstehung festhalten. In einem Konstruktionsverlauf werden viele konstruktive Entscheidungen getroffen, Lösungen erarbeitet und immer wieder verfeinert. Zudem werden einige Teilprozesse durch neue ersetzt (Leemhuis, 2005). Dieses Wissen und die dazugehörigen Erfahrungen gehören zum Produkt und können sehr wichtig und hilfreich für die zukünftigen Konstruktionen sein. Sie gehen jedoch oft verloren, ohne dokumentiert zu werden. Die prozessorientierte Abbildung bzw. Dokumentation des Konstruktionsprozesses und die damit gewonnene Transparenz des Ablaufs kann einen großen Beitrag zur Nachhaltigkeit leisten und dabei helfen, verborgene Optimierungspotentiale schneller zu identifizieren.

## 2.3 Modellbildung

Bevor die Modellierung von Prozessen detaillierter erörtert werden kann, soll zunächst kurz allgemein auf das Thema Modellbildung eingegangen werden. Zudem legt die Definition eines Modells die wichtigsten Eigenschaften der künftigen Abbildungen fest und soll demnach in Kapitel 2.3.1 einführend erläutert werden.

### 2.3.1 Modelldefinition

Ein Modell ist eine abstrakte Beschreibung eines Sachverhaltes (Patzak, 1982). Es dient einer Abbildung des Originals, anhand dessen sich zum Beispiel Prognosen über das Original leichter, billiger, schneller oder überhaupt erst treffen lassen (Stachowiak, 1973). Nach (Haun, 2013) zeichnet sich ein Modell durch Abstraktion aus und damit durch eine bewusste Vernachlässigung bestimmter Merkmale, um die für den Modellierer oder den Modellierungszweck wesentlichen Modelleigenschaften hervorzuheben. Die drei wesentlichen Charakteristiken eines Modells sind die Abbildung, die Verkürzung und der Pragmatismus (Stachowiak, 1973).

Abbildung bedeutet, dass ein Modell zur Darstellung von etwas natürlichem oder künstlichem dient. Verkürzung beschreibt die Tatsache, dass ein Modell im Vergleich zum Original nur eine bestimmte Menge der wichtigsten Merkmale beinhaltet und abhängig von den Anforderungen, die an das Modell gestellt sind (wie z. B. Genauigkeit und Ziele des Modelleinsatzes), variieren kann. Einschränkend können sich auch die zur Verfügung stehenden Modellierungswerkzeuge, der Abstraktionsgrad bzw. die Menge der abzubildenden Merkmale auswirken. Das letzte, pragmatische Merkmal soll für das Gleichgewicht des Nutzens und des Modellierungsaufwandes sorgen, denn Modelle werden oft personenbezogen erstellt bzw. nur für einen bestimmten Zweck, wobei die Lebenserwartung eines Modells auch begrenzt sein kann. (Kaiser, 2013)

### 2.3.2 Modellbildung in Konstruktion

Die Vielfalt der Entwicklungsmöglichkeiten technischer Systemen ist geprägt von der Vielzahl der Modellarten (Lindemann, 2009, Kaiser, 2013); angefangen von den abstrahierten Darstellungen in Form von Handskizzen bis hin zu Abbildung der komplexen Sachverhalte in Form von digitalen Schaltplänen und CAD-Modellen. Eines der Hauptziele jedes Modells, unabhängig von den Fachdisziplinen, ist das Abbilden des Sachverhaltes auf einer abstrakteren Ebene. Die Verwendung graphischer Modelle ermöglicht es vor allem, komplexe Sachverhalte für alle Beteiligten schneller begreifbar zu machen und eine Kommunikationsgrundlage zu schaffen (Kaiser, 2013, Ebert, 2013).

Gute Beispiele für visuelle Modelle sind die CAD-Geometrien, welche heute aus der Konstruktion nicht mehr wegzudenken sind. Durch die rasante Entwicklung der Rechentechnik haben dreidimensionale CAD-Modelle die Gestaltwahrnehmung im Vergleich zu den alten 2D-Zeichnungen gewaltig erhöht. Außerdem werden 3D-CAD Modelle heute noch für viele weitere Aufgabenstellungen wie zum Beispiel der Simulation verwendet. Die Notwendigkeit der Bildung realer Prototypen konnte damit beträchtlich reduziert und der Entwicklungsprozess beschleunigt und zeitlich verkürzt werden.

Insgesamt haben sich in den letzten Jahren Modellbildungen etabliert und auch modellbasierte Vorgehensweise und Methoden entstehen lassen wie das Model Based Systems Engineering oder der modellbasierte Systementwurf (Kaiser, 2013, VDI 2206:2004).

Im Rahmen der vorliegenden Arbeit steht jedoch eine andere Art des Modells im Fokus der Untersuchung, nämlich das Modell eines Prozesses, welches als „[...] Folge von Aktivitäten unter Nutzung von Information und Wissen sowie materiellen Ressourcen...“ (Lindemann, 2009) definiert wird, wobei die Aktivitäten für die Verarbeitung der Eingangsinformationen (Input) zu Ausgangsinformationen (Output) zuständig sind (Ponn und Lindemann, 2011).

### 2.3.3 Prozessmodelle

Prozessmodelle sind Abbildungen der Prozesse und dienen der Darstellung der Abfolge von einzelnen Aktivitäten oder Tätigkeiten in den Prozessen (siehe Abbildung 22). Meistens werden Prozessmodelle nach dem Top-Down-Prinzip aufgebaut, da sich oft aufgrund der Komplexität nicht alle Prozesse in einem Prozessmodell darstellen lassen. Dabei beginnt man auf der Ebene der Prozesse mit den Hauptaufgaben/Tätigkeiten und detailliert nach und nach die Teilaufgaben. Die Menge der Darstellungsebenen bzw. der sogenannten Sub-Prozesse ist vom Detaillierungsgrad abhängig. Die Verwendung von Prozessmodellen hilft, die vorhandenen Abläufe zu veranschaulichen und zu dokumentieren (BMI, 2016).





Abbildung 22: Schematische Darstellung eines Prozesses

Bei genügender operativer Detaillierung der Prozessmodelle können die Workflow-Modelle automatisiert erzeugt und Simulationen der modellierten Abläufe unmittelbar daraus abgeleitet werden.

In der folgenden Aufzählung werden die Hauptvorteile der Anwendung von Prozessmodellen genannt vgl. (BMI, 2016).

- Prozessmodelle erhöhen die Transparenz und Vermittlung des Verständnisses über die Abläufe, Teilschritte, Schnittstellen und verwendeten Ressourcen.
- Durch die Modellierung von Prozessen wird die Grundlage für die Schwachstellenanalyse bzw. Optimierung geschaffen.
- Prozessmodelle können als Grundlage für die Workflow-Modelle dienen, welche ihrerseits für die Entwicklung der Software und der elektronischen Datenverarbeitung notwendig sind.
- Aus organisatorischer Sicht sind die Prozessmodelle, als standardisierte Dokumentationsform für Abläufe, die Voraussetzung für die Zertifizierung nach DIN EN ISO 9001.

### 2.3.4 Grundlagen der Prozessmodellierung

Prozesse werden nach (Berkau, 1998, Gadatsch, 2012) in technische und betriebswirtschaftliche Geschäftsprozesse unterteilt (siehe Abbildung 23).

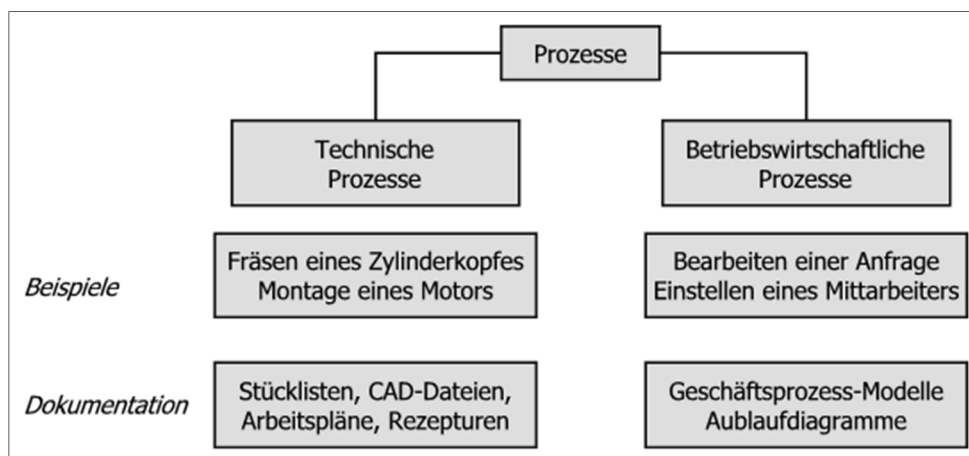


Abbildung 23: Unterteilung der Prozesse nach (Berkau, 1998)

Der Geschäftsprozess beschreibt „[...] eine zielgerichtete, zeitlich-logische Abfolge von Aufgaben, die arbeitsteilig von mehreren Organisationen oder Organisationseinheiten unter Nutzung von

Informations- und Kommunikationstechnologien ausgeführt werden können. Er dient der Erstellung von Leistungen entsprechend den vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen." (Gadatsch, 2012)

Unter einem technischen Prozess hingegen versteht man die Umformung und/oder den Transport von Materie, Energie und/oder Information und seine Zustandsgrößen, die mit technischen Mitteln gemessen, gesteuert und/oder geregelt werden können (DIN IEC 60050-351:2014-09). In der Abbildung 24 ist diese Definition des technischen Prozesses dargestellt.

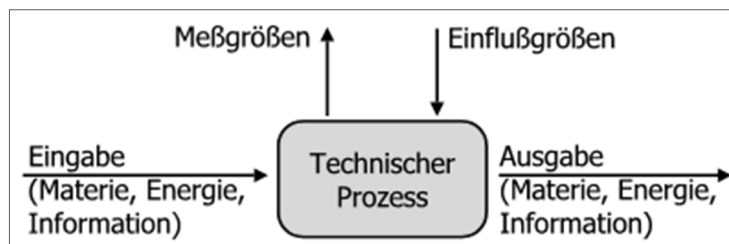


Abbildung 24: Veranschaulichung des Begriffes "technischer Prozess" (Hering, et al., 2000)

Im Rahmen der hier vorliegenden Arbeit sind zwar nur die technischen Prozesse von Interesse, die Grundlagen der Prozessverwaltung sind aber im kaufmännischen Bereich heute deutlich weiterentwickelt. Das Geschäftsprozessmanagement ist deshalb heute der Vorreiter und Know-How-Träger auf dem Gebiet des Prozessmanagements und hat einen großen Anteil an der praktischen Umsetzung entsprechender Methoden und Prinzipien.

Beim Modellieren von Prozessen werden generell drei Ebenen unterschieden: die strategische Ebene (Unternehmensziele und Strategien), die fachlich-konzeptuelle Ebene (Prozessmanagement) und die operative Ebene (Workflowmanagement).

Auf der strategischen Ebene erfolgt die Auseinandersetzung mit den Unternehmenszielen, welche anschließend die Grundlage für das Prozessmanagement bilden. Mit den so gesetzten Zielen und Strategien werden auf der fachlich-konzeptionellen Ebene die entsprechenden Prozesse entwickelt und modelliert (Gadatsch, 2012). Auf der operativen Ebene erfolgt schließlich die Anreicherung der Prozesse aus der fachlich-konzeptionellen Ebene mit weiteren technischen Details und Elementen. Dies ist notwendig, um den Prozess automatisieren und diesen mittels eines Workflow-Management-Systems ausführen zu können (Gehring und Gadatsch, 1999, Bösing und Geldmacher, 2014).

Unter einem Workflow-Management-System wird dabei ein Softwaresystem (Programm) verstanden, welches die Modellierung, die Ausführung und das Monitoring von Workflows unterstützt. Das System ist in der Lage die (semi-) formale Workflowbeschreibung zu interpretieren und auszuführen (Gadatsch, 2012). An dieser Stelle bietet es sich an, den Unterschied zwischen einem Prozess und einem Workflow kurz zu beschreiben und in einer entsprechenden Tabelle 1 gegenüberzustellen.

„Ein Workflow ist ein formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess. Er beinhaltet die zeitlichen, fachlichen und ressourcenbezogenen Spezifikationen, die für eine automatische Steuerung des Arbeitsablaufs auf der operativen Ebene erforderlich sind" (vgl. ebd). Mit anderen Worten ist ein Workflow die informationstechnische Realisierung eines Prozesses (Riggert, o. D.)

	Prozess	Workflow
Ziel	Analyse und Gestaltung der Arbeitsabläufe im Sinne gegebener strategischer Ziele	Spezifikation der technischen Ausführung von Arbeitsabläufen
Gestaltungsebene	Konzeptionelle Ebene mit Verbindungen zur Geschäftsstrategie	Operative Ebene mit Verbindungen zur unterstützenden Technologie
Detaillierungsgrad	Alle von einem Mitarbeiter an einem Arbeitsplatz ausführbaren Arbeitsschritte	Konkretisierung von Arbeitsschritten hinsichtlich der Arbeitsverfahren sowie personeller und technologischer Ressourcen

Tabelle 1: Gegenüberstellung Prozess und Workflow, adaptiert nach (Gadatsch, 2012)

Gut erkennen lässt sich, dass sich beide Begriffe zur Beschreibung von Arbeitsabläufen eignen. Der Unterschied liegt im Detaillierungsgrad und in der Nähe zur fachlichen Sicht (Prozess) bzw. zur technischen Umsetzung (Workflow) (Gadatsch, 2012). Die Abbildung 25 stellt diesen Unterschied schematisch dar.

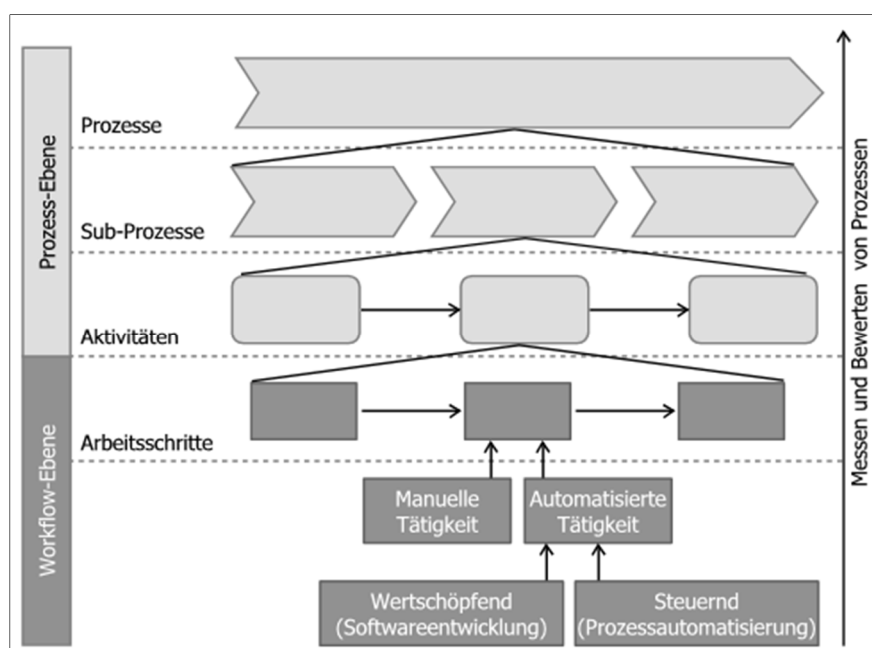


Abbildung 25: Prozess vs. Workflow (Freund und Götzer, 2008)

Die Grenze zwischen einem Prozess und dem entsprechenden Workflow verläuft genau zwischen den Aktivitäten und den elementaren Arbeitsschritten. Die Ausführung der elementaren Arbeitsschritte kann dabei manuell oder IT-gestützt erfolgen.

Die Möglichkeit der Ableitung eines Workflows aus einem Prozess bietet damit die Grundlage für die Automatisierung dieses Prozesses. Dabei soll die Methode, die für die Weitergabe der Informationen verwendet wird, formal sein. Die Formalität einer Methode wird grundsätzlich durch zwei Aspekte definiert. Zum einen durch die formale Semantik, welche den zu verwendenden Zeichensatz beschreibt und zum anderen durch die eindeutigen Regeln für die Interpretation dieses Zeichensatzes (Syntax), welche beim Sender und Empfänger gleich sein sollen. Beim Modellieren der Berechnungsprozesse müssen diese beiden Voraussetzungen erfüllt sein, da ein Berechnungsprozess im Prinzip eine Sammlung von Informationen darstellt, welche vom Modellierer (Sender) an das Programm (Empfänger) zur Automatisierung des Prozesses weitergegeben wird (Bösing und Geldmacher, 2014).

## 2.4 Prozessautomatisierung

Das Thema Prozessautomatisierung hat mittlerweile eine weit bis in das 18. Jahrhundert zurückreichende Geschichte,. Zu diesem Zeitpunkt umfasste sie vor allem die manuellen Tätigkeiten, bei der die größeren Fertigungsprozesse in kleinere, unabhängige Aufgaben aufgesplittet wurden, die ihrerseits von spezialisierten Arbeitern durchgeführt werden konnten. Die Mitarbeiter mussten lediglich lernen, eine einzige einfache Aufgabe zu beherrschen, ohne dabei die gesamte Prozesskette kennen zu müssen. Der Vorteil dieses Ansatzes waren für das Unternehmen die wesentlich geringeren Kosten für Produktion und Personal aufgrund der Einfachheit der Aufgaben und dessen höheren Automatisierungsgrad. Mit der Weiterentwicklung der Technologien verstärkte sich dieser Trend. Immer mehr manuelle Aufgaben wurden durch Maschinen und nachfolgend durch hocheffiziente Roboter erledigt. Außerdem wurden auch viele kaufmännische Unternehmensfunktionen von der Bestellung der benötigten Rohmaterialien über Logistik bis hin zur Aufrechterhaltung der Kundenbeziehungen unter Verwendung komplexer Softwaresysteme automatisiert. Als Ergebnisse konnten über die verkürzten Zeiten bis hin zur Markteinführung neuer Produkte auch die Verringerung der Herstellungskosten beobachtet werden. Das eröffnete den Unternehmen völlig neue Potentiale und Möglichkeiten sowohl national als auch international. Die stetig steigende Produkt- und Prozesskomplexität sowie eine Vielzahl von verschiedenen beteiligten Systemen forderten darüber hinaus eine Standardisierung, die zur Entstehung eines neuen Gebietes in Forschung und Anwendung führte, das sowohl die Betriebswirtschaft als auch Informationstechnik in sich vereint. (Petkov, 2014).

Die Grundidee hinter diesem Business Process Management ist die Bereitstellung eines strukturierten Weges für das Entwerfen, Ausführen und Überwachen von verschiedenen Prozessen, welche der

Erreichung eines bestimmten Ziels dienen (van der Aalst, et al., 2003). Durch den Ursprung im Bereich der Unternehmensführung und Organisation stammt ein großer Teil der Terminologie aus dem betriebswirtschaftlichen Bereich. Die Grundeinheit von Business Process Management ist ein Geschäftsprozess. Es wurde ursprünglich als eine Sammlung von Aktivitäten definiert, die bestimmte Eingaben erfordern, um gewünschte Ausgaben zu produzieren, welche einem bestimmten Zweck dienen (Hammer und Champy, 1988).

Auch wenn ein Geschäftsprozessmodell sehr detailliert sein kann (siehe Kapitel 2.4.4), seine Hauptidee ist, die zugrunde liegenden Aktivitäten und deren Korrelationen zu veranschaulichen. Die Informationen über die spezifische Ausführung des Modells auf einem bestimmten Software-System zu beschreiben erfolgt hingegen mit den Werkzeugen und Methoden aus dem Bereich der Informationstechnik auf der operativen Ebene. Hier wird der Prozess in einen Workflow überführt, um dann vom Rechner „versandt“ und ausgeführt zu werden (Petkov, 2014).

Die Automatisierung von Konstruktionsprozessen mit Hilfe ausführbarer Workflows stellt eines der Ziele der vorliegenden Arbeit dar. Konstruktionsautomatisierung bedeutet allerdings keineswegs, dass hier von der Anforderungsdefinition bis hin zur Fertigung vollautomatische Prozesse ablaufen. Die Automatisierung soll vor allem auf der operativen Ebene stattfinden. Die Kreativität und das Wissen der Ingenieure verlagern sich andererseits auf die fachlich konzeptionelle Ebene der Modellierung der Konstruktionsabläufe und der Entwicklung und den Ausbau von Wissens-Datenbanken.

Um zumindest eine Teilautomatisierung der Prozesse zu ermöglichen, bedarf es einer formalen Sprache zur Beschreibung der Abläufe. Die formale Beschreibungssprache ist die Voraussetzung für eine rechnergestützte Modellierung mit anschließender Ausführung (Simulation) der Prozesse (Rüppel, 2007). Im Laufe der Zeit sind viele verschiedene Sprachen und Standards zur Beschreibung von Geschäftsprozessen entstanden. Einige von ihnen bieten rein textbasierte XML-Spezifikationen ohne grafische Notationen. Andere, wie zum Beispiel die Unified Modeling Language (UML), unterstützen zwar die grafische Modellierung, fanden bis jetzt aber keine breite Akzeptanz in der technischen Anwendung. Ein kurzer Überblick über die wichtigsten Standards für die Modellierung von Geschäftsprozessen wird in den folgenden Kapiteln gegeben.

#### 2.4.1 Petri-Netze

Petri-Netze gelten allgemein als Ursprung für Simulationstechnik und Verfahrensabbildung (Bösing und Geldmacher, 2014). Sie wurden entwickelt, um das Verhalten der informationsverarbeitenden Automaten zu modellieren (Freund und Götzer, 2008). In seiner einfachsten Form (siehe Abbildung 27), ist ein Petri-Netz ein gerichteter Graph, welcher nur aus zwei verschiedenen Arten von Knoten besteht: Stellen (auch Bedingung oder Zustand genannt) und Transitionen (Ereignis). Eine Stelle wird durch einen Kreis dargestellt und symbolisiert eine Ablage für Objekte oder Daten. Transitionen

beschreiben die Verarbeitung von Objekten und werden durch einen Balken (Rechteck) dargestellt. Gerichtete Kanten (Pfeile) geben die logische Folge an und dürfen nur von Knoten der einen Sorte zu Knoten der anderen Sorte führen. Die Stellen, zu denen eine Kante von einer Transition  $t$  führt, heißen Ausgabestellen von  $t$ , Stellen von denen eine Kante zu der Transition  $t$  führt heißen Eingabestellen von  $t$ . Das Verarbeiten einer Transition heißt auch Schalten (Schwill, o.D.), (Freund und Götzer, 2008). Mit Hilfe von Token (dt. Marken) kann die Dynamik in den Petri-Netzen realisiert werden. Die dahinterstehende Logik ist relativ einfach, trotzdem lassen sich damit auch komplexe Sachverhalte beschreiben (Freund und Götzer, 2008). Die Grundelemente der Petri-Netze sind in der Abbildung 26 dargestellt.



Abbildung 26: Elemente der Petri-Netze

Die Dynamik der Petri-Netze wird durch das Schalten (Verarbeiten) von Transitionen abgebildet. Die Regel für das Schalten lautet wie folgt: eine Transition schaltet nur dann, wenn alle Eingabestellen der Transition markiert sind und alle Ausgabestellen leer sind. Die Markierung einer Stelle erfolgt mit den Token (Marke) (Freund und Götzer, 2008, Schwill, o.D.). Durch die Analogie zu einer mathematischen Funktion kann das Schalten folgendermaßen formuliert werden: eine Funktion kann nur dann ausgeführt werden, wenn alle notwendigen Eingangsdaten (Parameter) vorliegen und das Ergebnis nicht. Ein einfaches Beispiel, womit sich die Logik einfach veranschaulichen lässt, ist in der Abbildung 27 dargestellt.

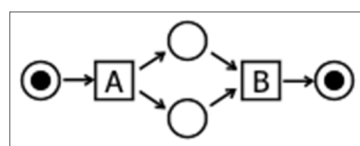


Abbildung 27: Beispiel eines Petri-Netzes (Freund und Götzer, 2008)

Das Ereignis „A“ kann schalten, da die Vorbedingung erfüllt (weil markiert) ist, und beide Nachbedingungen nicht. Ereignis „B“ kann nicht schalten, da die Vorbedingungen nicht erfüllt sind, die Nachbedingung jedoch schon (vgl. ebd.). Mit den Petri-Netzen lassen sich im Prinzip nur Ablauflogiken modellieren, genauere Informationen zum Prozess, wie Informationsobjekte oder Ressourcen, können nur über die textuelle Beschriftung angegeben werden. Damit ist eine reine Petri-Netzmodellierung in der Prozessmodellierungspraxis eher selten vorzufinden. Außerdem erfordern komplizierte Abläufe in ihrer Gesamtheit sehr große Graphen und werden auch dementsprechend schnell unübersichtlich (Freund und Götzer, 2008).

Ein weiterer Nachteil von Petri-Netzen liegt in der Modellierung von logischen Operatoren, wie XOR, OR oder AND, denn diese können in Petri-Netzen nicht eindeutig implementiert werden. Dies ist auch einer der Gründe dafür, dass Petri-Netze erweitert werden, wobei sie aber ihre Formalität teilweise verlieren (Bösing und Geldmacher, 2014).

## 2.4.2 UML - Aktivitätsdiagramme

Eine weitere Diagrammart, um Abläufe zu modellieren, sind Aktivitätsdiagramme der Unified Modeling Language (UML). Mit ihrer Hilfe können auch komplexe Verläufe unter der Berücksichtigung von Nebenläufigkeit, alternativen Entscheidungswegen und ähnlichem abgebildet werden (Rupp, et al., 2012).

Eine Aktivität im Sinne der UML 2.0 spezifiziert dabei eine Menge von potentiellen Abläufen, die sich in der Realität („zur Laufzeit“) unter bestimmten Randbedingungen abspielen (ebd.). Ein Aktivitätsdiagramm kann auch als gerichteter Graph verstanden werden. Die Grundelemente eines Aktivitätsdiagramms sind die Aktivitäten, Aktionen, Objektknoten, Kontrollelemente und verbindende Kanten (siehe Abbildung 28). Eine Aktion beschreibt dabei einen elementaren Einzelschritt (Arbeitsschritt), welcher den Ablauf durchschreitet. Objektknoten repräsentieren die beteiligten Daten oder Gegenstände. Zwischen den Aktionen befinden sich die Kontrollknoten zur Steuerung des Ablaufs. Diese geben Entscheidungsregeln bzw. Bedingungen vor, in welcher Reihenfolge und wann die einzelnen Aktionen ausgeführt werden. Durch Kontrollknoten lassen sich die Abläufe zum Beispiel parallelisieren, zusammenführen, verzweigen oder auch abbrechen. Die Kanten verbinden die Diagrammelemente und damit die zeitliche und logische Reihenfolge des Ablaufs.

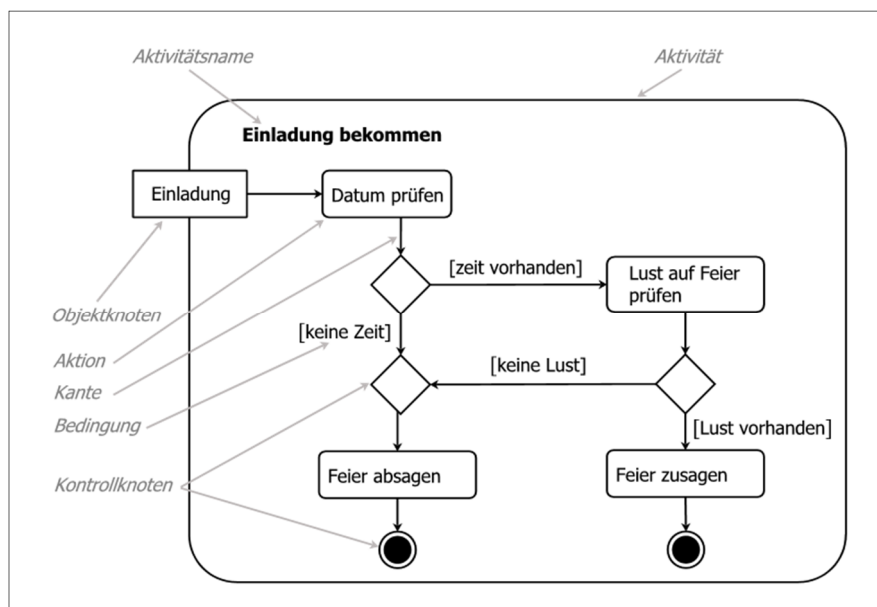


Abbildung 28: Grundelemente einer Aktivität, adaptiert nach (Rupp, et al., 2012)

Zur Realisierung der Nebenläufigkeit in den Abläufen wird das sogenannte Token-Konzept verwendet (Hitz, et al., 2005). Im Prinzip ähnelt dieses Konzept den Schaltregeln von Petri-Netzen. Außer, dass die Token in den Aktivitätsdiagrammen nicht explizit dargestellt werden und nur zur logischen Erklärung dienen. Die Wanderung eines Tokens durch eine Aktivität repräsentiert dabei ihre Abarbeitung (Rupp, et al., 2012).

Eine Aktion wird von einem Token ausgelöst. Eine Aktion startet dann, wenn ein Token sie über eine Kante erreicht. Dieses Token wird anschließend für die Dauer der Abarbeitung der Aktion entfernt und aufbewahrt. Ist die Abarbeitung der Aktion abgeschlossen wird das Token über die ausgehende Kante zur nachfolgenden Aktion weitergeschickt (siehe Abbildung 29) (vgl. ebd.)

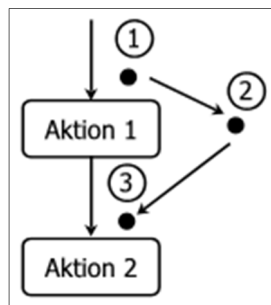


Abbildung 29: Darstellung der Abarbeitung einer Aktion nach dem Token-Prinzip

Nach vorgenanntem Prinzip können sequentielle sowie parallele Abfolgen von Aktivitäten modelliert werden. Wenn aber ein Prozess eine Vielzahl von parallelen Aktivitäten aufweist, wird das Modell aufgrund der zahlreichen Synchronisationsknoten rasch unübersichtlich. Mit Hilfe sogenannter Pools (dt. Partitionen) können in UML bestimmte Kanten und Knoten auf Basis gemeinsamer Eigenschaften gruppiert werden. In Abbildung 30 ist ein einfaches Beispiel für eine Terminkoordination mit Anwendung von Pools dargestellt. Dabei sind die Aktionen „Teilnehmer auswählen“ und „Eckdaten des Termins erfassen“ mit Hilfe eines Pools dem Benutzer und solche wie „Benutzer ID prüfen“ oder „Kollision prüfen“ dem System Kalendarium zugeordnet.



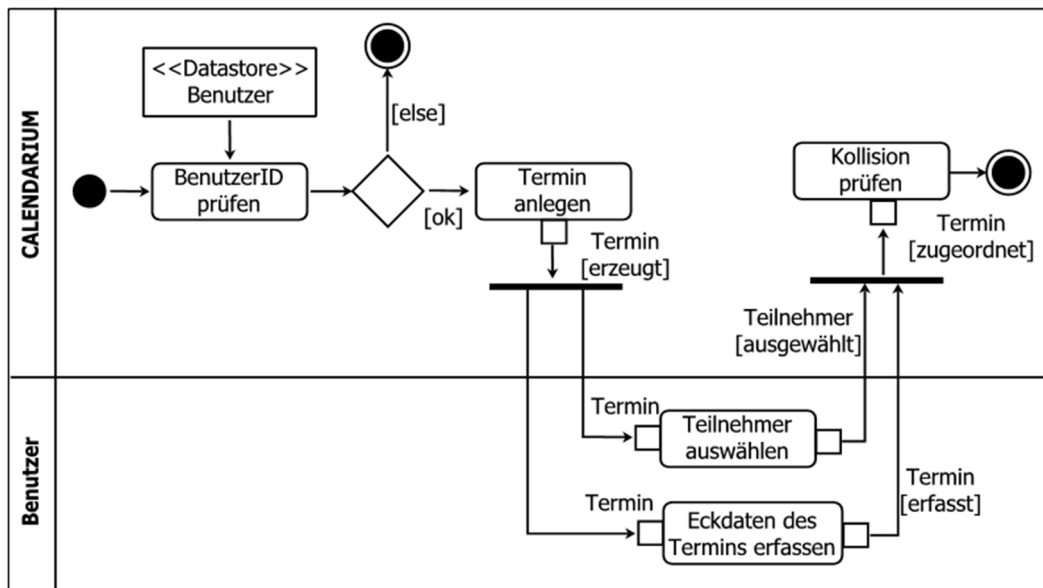


Abbildung 30: Beispiel der Anwendung von Pools, adaptiert nach (Hitz, et al., 2005)

Mit Hilfe von Pools können somit zwar Prozesse in verschiedene Aktivitäts- bzw. Verantwortungsbereiche unterteilt werden, die Modellierung der komplexen Abläufe bleibt jedoch aufwendig (Bösing und Geldmacher, 2014), nicht zuletzt auch durch die Fülle an Notationselementen (Rupp, et al., 2012). Die Aktivitätsdiagramme können abhängig vom abzubildenden Verhalten sehr komplex werden, beschreiben dafür das Verhalten aber sehr präzise (Freund und Götzer, 2008).

Damit der Datentransfer beziehungsweise der Objektfluss bei der Verhaltensmodellierung abgebildet werden kann, existieren in der UML die sogenannten Objektknoten bzw. die Pins. Diese Modellierungselemente können Informationen aufnehmen und diese der Aktivität beziehungsweise einer Aktion zur Verfügung stellen. Außerdem können diese auch gleichzeitig die Ergebnisse einer vorangegangenen Aktion sein. In der Abbildung 30 wird zum Beispiel der Termin als Pin von der Aktion „Termin anlegen“ an die Aktionen „Teilnehmer auswählen“ und „Eckdaten des Termins erfassen“ übergeben. Der Unterschied zwischen dem Objektknoten und einem Pin liegt lediglich darin, an welchem Modellelement diese „angeheftet“ sind. Der Pin ist ein formaler Parameter einer Aktion, wohingegen es sich bei den Objektknoten um die Eingabe- und Ausgabeparameter einer Aktivität handelt. Grafisch wird ein Objektknoten in Form eines Rechteckes überlappend an einer Aktivität dargestellt. Ein Pin wird dagegen nicht überlappend an einer Aktion dargestellt (siehe Abbildung 31).

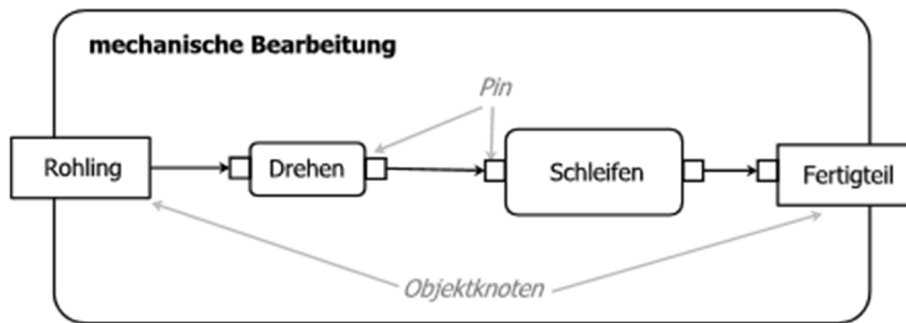


Abbildung 31: Beispiel des Informationsflusses in Aktivitätsdiagrammen

Ein Informationstransfer erfolgt dabei nur dann, wenn der Typ des Objektknotens/Pins des Senders und des Empfängers übereinstimmen (Hitz, et al., 2005). Die Typen der Objektknoten selbst können in UML-Aktivitätsdiagrammen völlig frei definiert werden und damit alle möglichen Daten und Objekte repräsentieren. Außerdem unterstützt das Aktivitätsdiagramm die Modellierung von nicht objektorientierten Systemen, womit die Aktivitäten unabhängig von Objekten definiert werden können (ebd.). Dies könnte zum Beispiel sehr hilfreich beim Aufbau von Funktionsbibliotheken sein.

### 2.4.3 Ereignisgesteuerte Prozessketten

Die Ereignisgesteuerte Prozesskette (EPK) ist eine semi-formale graphische Modellierungssprache, die von Scheer entwickelt wurde, um sachlogische Zusammenhänge von Geschäftsprozessen in einem Unternehmen einfach und möglichst realitätsnah abbilden zu können (Scheer, 2002).

Diese Methodik knüpft an die Petri-Netze an und erweitert sie in pragmatischer Art und Weise (Freund und Götzer, 2008). In ihrer Grundform sind die EPK's gerichtete bipartite<sup>1)</sup> Graphen, welche aus folgenden Elementen bestehen:

- Funktion (Ausführung einer Tätigkeit)
- Ereignisse (Auslöser bzw. das Ergebnis einer Tätigkeit)
- Konnektoren (Verknüpfungsoperatoren wie zum Beispiel ODER bzw. UND)
- Kanten (Daten- oder Kontrollflussangabe)

Eine Bipartität des Graphen beschreibt dabei die Modellierungsregeln: es ist strengstens gefordert, dass sich die Funktionen und Ereignisse abwechseln und somit eine Kante nur unterschiedliche Knotentypen verbindet. Zusätzlich zu der direkten Art der Verbindung mittels einer Kante können Funktionen und Ereignisse über Verknüpfungsoperatoren miteinander verbunden werden. Dadurch können die Kanten verzweigen und somit die Nebenläufigkeit der Abfolgen modelliert werden (Freund und Götzer, 2008, Bartsch, 2010). Ein einfaches Beispiel einer ereignisgesteuerten Prozesskette für den Vorgang zur Behebung einer Störung ist in der Abbildung 32 dargestellt.

1) Ein Graph ist bipartit, wenn seine Knotenmenge sich in zwei disjunkte Teilmengen zerlegen lässt, sodass keine seiner Kanten zwei Knoten einer Teilmenge verbindet (Aigner, 2015).

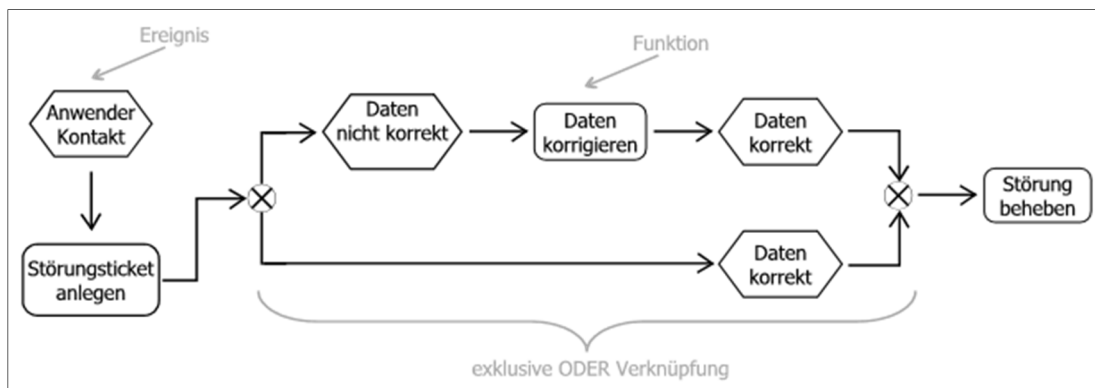


Abbildung 32: EPK Beispiel (Bartsch, 2010)

Um weitere Aspekte der realen Welt abbilden zu können, wurden die EPK's erweitert und damit die erweiterten EPK's geschaffen. Diese beinhalten zusätzliche Elemente wie zum Beispiel Rollen, Informationsobjekte oder Organisationseinheiten. Dies ist auch gleichzeitig eine der Schwächen von EPK's, denn mit wachsender Menge an ergänzenden Informationen werden die EPK's schnell unübersichtlich, so dass der Prozessfluss schwer erkennbar ist (Freund und Götzer, 2008). Außerdem sind im Vergleich zu den formalen Sprachen, wie z. B. den Petri-Netzen, weder Syntax noch Semantik präzise definiert, was ein großes Hindernis aus Sicht der Automatisierung bzw. der Workflow-Orientierung darstellt (Bartsch, 2010).

#### 2.4.4 Business Process Management und Notation (BPMN)

Business Process Management (BPM) ist ein relativ neuer Ansatz im Geschäfts- und Prozess-Reengineering, der erst mit der Einführung der objektorientierten Modellierung in der Software-Programmierung Fahrt aufnahm. Inzwischen hat sich diese Methode vor allem in betriebswirtschaftlich orientierten Geschäftsprozessen zur Modellierung und Ausführung von Programmen bewährt (Gründer und Polyakov, 2011). Dabei liegen die Schwerpunkte der bislang automatisierten Prozesse in der Beschleunigung von Kommunikationsaufgaben, der Vermeidung von Wiederholktivitäten im Datentransfer zwischen verschiedenen Softwareanwendungen und in der autonomen Formularbearbeitung (Polyakov und Gründer, 2013).

Die Geschäftsprozesse, die mit Hilfe der Business Process Modeling Notation (BPMN) definiert werden, können sequentiell oder parallel ablaufen und menschliche Aktivitäten ebenso wie IT-Prozesse beinhalten. Eine der wichtigsten Funktionen von BPMN ist neben der Analyse und Strukturierung von Prozessen ihre Umsetzung in rechnerunterstützte Abläufe und die Erstellung von Benutzersystemen. Ziel hierbei ist es, die nahtlose Kombination von manuellen und intellektuellen Tätigkeiten mit IT-Anwendungen und Kommunikationsaufgaben sowohl am Arbeitsplatz wie zwischen Abteilungen, Teams und Einzelpersonen in verteilten Entwicklungsprojekten zu garantieren.

Die einmal interaktiv modellierte Workflow-Logik kann in lauffähige Programme umgewandelt werden (Gründer und Polyakov, 2011). Diese wiederum verlinken die Benutzeroberflächen mit den Mail-Systemen, Datenbanken, Anwendungen, Informationsquellen, Event-Handlern und der Projektleitung. Die Verlinkung erfolgt über eine flexible Kommunikationsinfrastruktur. Traditionelle Verfahren, Arbeitsmethoden und Gewohnheiten können mit dieser Technologie in einer sehr flexiblen Umgebung verbunden werden.

Die Abläufe lassen sich in BPMN mit Hilfe von Prozessdiagrammen definieren und darstellen. Es gibt vier unterschiedlichen Arten von Modellelementen (OMG, 2011):

- Fluss-Objekte (flow objects)
- Verbindungsobjekte (connecting objects)
- Schwimmbahnen (swimlanes) und
- Artefakte (artefacts)

In der Abbildung 33 sind die Basiselemente für die Definition eines Ablaufes nach der BPMN – Terminologie aufgeführt. Dazu gehören das Ereignis als Kreis dargestellt, die Aktivität als Rechteck mit abgerundeten Kanten und das Gateway (dt. Entscheidung) als Raute (Siegel, 2008).



Abbildung 33: Grundelemente von BPMN

Die Prozessbeteiligten werden mit Hilfe von Schwimmbahnen angegeben. Ein einfacher Beispielprozess für eine Lizenzanforderung einer neuen Software in einem Unternehmen ist mit den Grundelementen in der Abbildung 34 dargestellt.

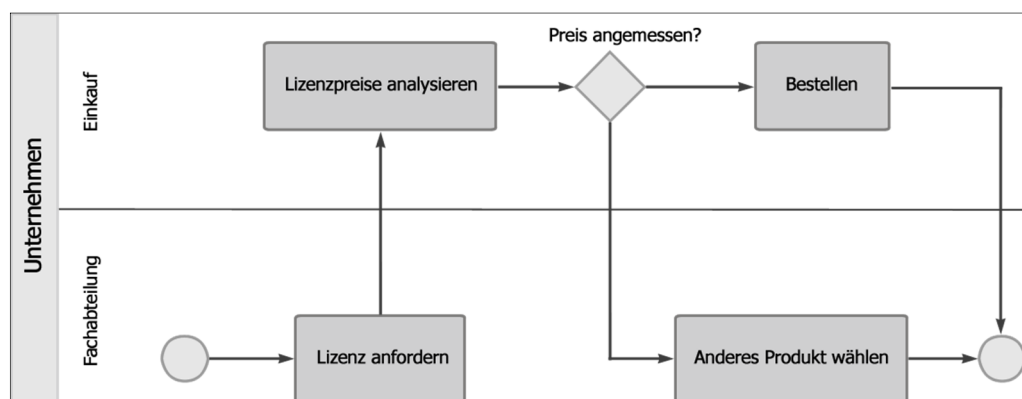


Abbildung 34: Beispiel eines BPMN-Prozessdiagramms (Götz, 2011)

Ein Unternehmen wird in diesem Beispiel als Pool mit zwei Schwimmbahnen dargestellt, welche verschiedene Prozessteilnehmer (Einkaufsabteilung und eine Fachabteilung) repräsentieren sollen. Jedem Prozessteilnehmer sind nur bestimmte Aktivitäten zugewiesen, die ausgeführt werden sollen. Somit ist die Einkaufsabteilung zum Beispiel für die Analyse der Lizenzpreise zuständig und nicht die Fachabteilung. Im Gegensatz dazu kann nur die Fachabteilung im Fall eines zunächst zu teuer angebotenen Produktes eine Alternative aussuchen und nicht der Einkauf. Die Sequenz-Flüsse (als Pfeile dargestellt) geben dabei die Richtung des Ablaufes an (vgl. ebd.). Da die Geschäftsprozesse nur in den seltensten Fällen sequenziell ablaufen, gibt es in der BPMN das Element Gateway, mit dessen Hilfe die Aufspaltung und die Zusammenführung eines Kontrollflusses modelliert werden können. In dem beschriebenen Beispiel ist dieses Element für die Entscheidung zuständig, ob der Preis der angeforderten Lizenz aus der Sicht des Unternehmens angemessen ist oder nicht. Abhängig von der Entscheidung leitet das modellierte XOR-Gateway, bei dem entweder der eine oder der andere Pfad ausgeführt werden kann, den Kontrollfluss weiter zur Bestellung oder zur Auswahl einer Alternative (Götz, 2011).

Zusätzlich zu einer elementaren Aktivität existiert in der BPM Notation eine zusammengesetzte Aktivität (SUB-Prozess). Dieses Element ermöglicht eine hierarchische Darstellung der Prozesse (siehe Abbildung 35). Die Darstellung erfolgt wie bei einer Aktivität. Zusätzlich ist im unteren Bereich des Symbols ein „+“ Zeichen zum Öffnen des Subprozesses platziert. Ein Subprozess kann wiederum ein Prozessdiagramm mit allen bisher vorgestellten Elementen enthalten. Dieses Diagramm beschreibt, was bei der Ausführung von diesem Subprozess geschehen soll. Damit hat ein Subprozess keine eigentliche Bedeutung für die Lenkung des Prozesses, sondern ermöglicht eine kompaktere hierarchische Strukturierung von Prozessmodellen.

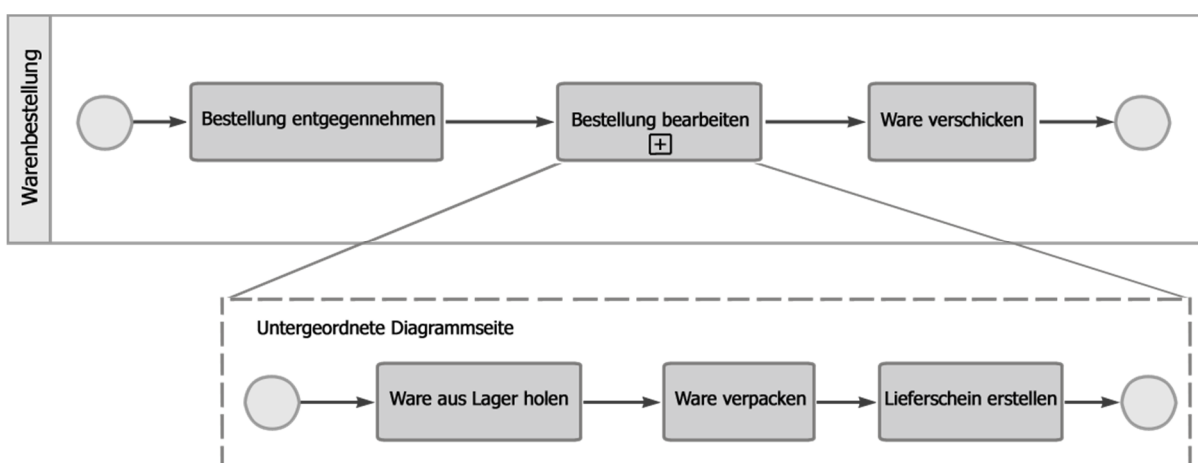


Abbildung 35: Hierarchie durch SUB-Prozesse

Im Gegensatz zur UML richtet sich die BPMN an einen weitaus größeren Kreis von Anwendern. Da sie vergleichsweise verständlich und einfach ist, kann sie nicht nur von den Systemanalytikern, die die fachlichen Modelle erstellen, sondern auch von den Entwicklern gelesen werden, die für die erstellten

Modelle die Prozessausführung implementieren. Die Notation besitzt auch deshalb eine hohe Akzeptanz, weil die Grundelemente der BPMN den bekannten Flussdiagrammen ähneln, und es nur eine Diagrammart gibt. Die Verwendung von Swimlanes und Sub-Prozessen strukturiert die Prozesse in einer Weise, die auch komplexe Abläufe mit einer hohen Transparenz ausstattet. Sofern die modellierten Prozesse mit entsprechenden Werkzeugen in dem dafür vorgesehenen XML-Format (BPEL) abgespeichert werden, können die Prozesse anschließend automatisch ausgeführt werden. Zu den Nachteilen der Notation zählen dagegen die starke Ausrichtung auf die Geschäftsprozessmodellierung sowie die fehlende Möglichkeit, Datenmodelle und Systemarchitekturen beschreiben zu können. Eine Kombination der beiden Notationen UML und BPMN zu einer verständlichen visuellen Sprache hätte das Potential, sowohl das System mit seinen Prozessen, als auch dessen Verhalten konsistent zu beschreiben (M.-V., 2011).

Im Allgemeinen ähneln die in BPMN verwendeten Diagramme allesamt dem Aktivitätsdiagramm der UML. Bei dem letzten wurden auch Konzepte wie „Activity“ und „Swimlane“ übernommen. Die Lesbarkeit wurde hingegen durch die Einführung von zahlreichen weiteren Elementen gegenüber einem Aktivitätsdiagramm deutlich verbessert. Trotz der Erweiterbarkeit der Spezifikation bleibt BPMN dennoch stark auf Geschäftsprozesse ausgerichtet. Im Fall einer modellbasierten Vernetzung müssten die produktspezifischen Informationen den anderen Modellen entnommen werden (Zingel, 2013).

## 2.5 Systems Engineering (SE)

Die Produktentwicklung hat sich die möglichen Vorteile der Workflow-Nutzung bisher nur in Ansätzen erschlossen. Vielmehr konzentrierte man sich hier auf die Methoden und Vorgehensweisen für die Problemlösung und Komplexitätsbeherrschung. Die im Kapitel 2.2 beschriebenen phasenorientierten integralen Konstruktionsmethoden haben zwar für das Freigabewesen und andere standardisierte Prozesse Werkzeuge hervorgebracht, den Gesamtprozess der Produktentstehung unterstützen sie dagegen nicht.

Der Systems Engineering Ansatz, der in seinen Grundzügen bereits in die VDI 2206:2004 eingeflossen ist, bietet in seiner Weiterentwicklung gute Voraussetzungen für eine prozessintegrierte Weiterentwicklung der heutigen Konstruktionsmethoden. Die besondere Bedeutung erhält das Systems Engineering (SE) in der Produktentwicklung, da es die Problemlösungsprozesse bei komplexen Problemstellungen methodisch unterstützen kann. Zu den wesentlichen Methoden beim Systems Engineering gehören die Systems Engineering - Philosophie und der Problemlösungsprozess. Die Methode des Systems Engineering konzentriert sich nach (Weilkiens, 2006, oose., o.D.) auf die folgenden Methodenschritte:

- die Dekomposition eines Systems in Subsysteme,
- die Dokumentation der Systemanforderungen in der frühen Entwicklungsphase,
- die Erarbeitung eines Systemdesigns und
- die Überprüfung des Systems auf Einhaltung der gestellten Anforderungen.

Letzteres geschieht unter Berücksichtigung des Gesamtproblems in Hinblick auf Betrieb, Zeit, Tests, Kosten und Planung sowie Entsorgung. Das Systems Engineering ermöglicht eine gleichberechtigte Integration aller Fachdisziplinen (wie bspw. Maschinenbau, Elektrotechnik, Informationstechnik, etc.) und die Bildung eines strukturierten Entwicklungsprozesses von der Idee bis zur Betriebsphase, wobei nicht nur die technischen, sondern auch die wirtschaftlichen Aspekte einbezogen werden, um schließlich ein benutzergerechtes System zu entwickeln. Der Ursprung des Systems Engineering liegt in der Systemtheorie. Durch den stetigen Zuwachs der Produktkomplexität gewann der systemorientierte Ansatz immer mehr an Bedeutung. Heute wird er besonders durch die Forschungen zu den Themen Industrie 4.0 und Cyber-Physical Systems vorangetrieben (Gausemeier, et al., 2013).

Die SE-Philosophie ist ihrerseits durch das Systemdenken und das eigene Vorgehensmodell geprägt, wobei der Problemlösungsprozess die Systemgestaltung, in der die eigentliche konstruktive Lösungsfindung erfolgt, und das Projektmanagement, das für die Koordination und Organisation des Problemlösungsprozesses zuständig ist, unterstützt (Baumberger, 2007, Kaiser, 2013) (siehe Abbildung 36).

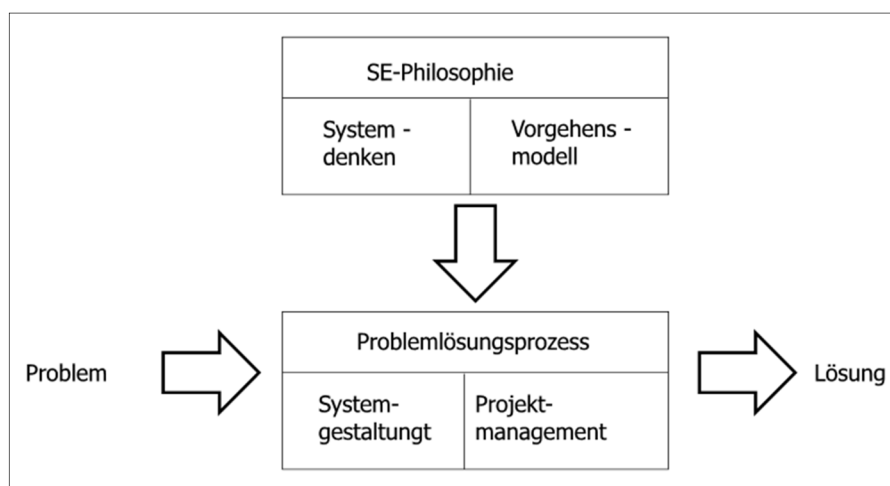


Abbildung 36: Die wesentlichen Bestandteile von Systems Engineering, adaptiert nach Daenzer und Huber, 1999

Unter dem Systemdenken ist vor allem eine ganzheitliche und systematische Denkweise bei der Betrachtung komplexer Systeme und deren Bestandteile zu verstehen. Die Betrachtung aus verschiedenen Perspektiven und Blickwinkeln hilft dabei, in Bezug auf die konkrete Problemstellung

die nötigen Zusammenhänge zu erschließen (Baumberger, 2007). Die Dokumentation der Systemmodelle reduziert dabei vor allem die Gefahr, wichtige Details nicht zu berücksichtigen (Spahni, 2000). Die bildhafte Darstellung des Systemmodells ist hierbei von ausschlaggebender Bedeutung. Das SE-Vorgehensmodell sieht ein Top-Down-Vorgehen (vom Groben zum Detail) vor, wodurch zuerst die generellen Ziele für das gesamte System und der entsprechende Lösungsraum festgelegt werden, und erst dann die Konkretisierung und Detaillierung der einzelnen Lösungskonzepte erfolgen soll. Die Variantenbildung soll dafür sorgen, dass möglichst viele Lösungsprinzipien in Betracht gezogen werden und nicht nur das erstbeste eine Beachtung findet (Spahni, 2000). Der Problemlösungszyklus stellt in dem Systems Engineering-Vorgehensmodell einen generellen Leitfaden zur Lösung von unterschiedlichen Problemen dar (Acél, 1996) (siehe Abbildung 37). Ein entsprechender Phasenablauf hilft dabei, den Entwicklungsprozess mit Planungs-, Konkretisierungs- und Entscheidungsmeilensteinen sowie den Werdegang einer möglichen Lösung zeitlich und ressourcentechnisch zu organisieren (Spahni, 2000).

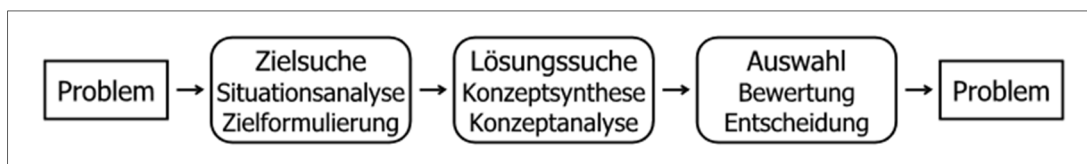


Abbildung 37: Problemlösungszyklus im Systems Engineering (Acél, 1996, Daenzer und Huber, 2002)

Die VDI-Richtlinie 2206:2004 hat mit dem Einfließen des V-Modells in den Kontext der Richtlinie die Vorgehensweise des Systems Engineering Ansatz bestätigt (Spahni, 2000).

### 2.5.1 Modellbasiertes Systems Engineering (MBSE)

Während klassische Methoden des Systems Engineerings dokumentenorientiert sind, bildet beim Modellbasierten Systems Engineering (MBSE) ein digitales Systemmodell die Basis, da digitale Modelle eine einfachere Komplexitätserfassung sowie einen Informationsaustausch zwischen den verschiedenen Disziplinen erlauben. Die während der Produktentwicklung entstehende abstrakte modellbasierte Spezifikation des Produktes ist für alle an der Entwicklung Beteiligten verständlich. Eine frühzeitige Definition der Korrelation zwischen den Systemanforderungen, Funktionen, Strukturen und dem Verhalten sowie deren Abbildung im Systemmodell kann später die Integration der einzelnen Systemkomponenten deutlich vereinfachen (Eigner, et al., 2012).

Der Hauptunterschied zum klassischen Systems Engineering besteht darin, dass hier als Ziel die vollständige Dokumentation des Systems verfolgt wird: Denn beim MBSE wird die Kommunikation der Systemkomponenten untereinander durch ein virtuelles Modell des kompletten Systems dokumentiert. Das bedeutet, dass beim MBSE die Änderungen am Modell inklusive aller daraus resultierender Einflüsse sofort sichtbar werden (siehe Abbildung 38).



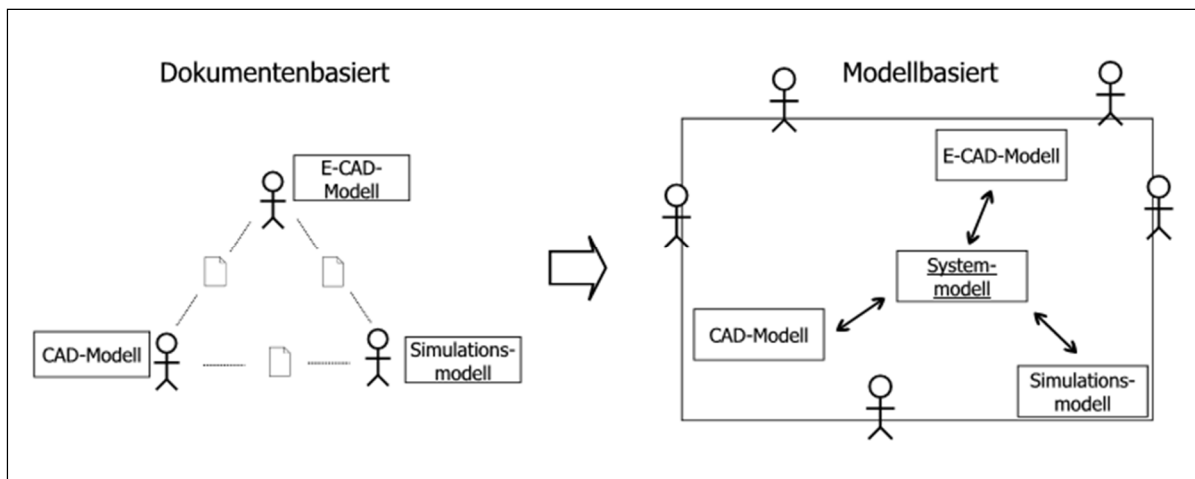


Abbildung 38: Dokumentenbasiert vs. Modellbasiert (Eigner, et al., 2014)

Eine konsistente Datenbasis und eine formale Beschreibungssprache für die Modelle bilden die Voraussetzungen für ein erfolgreiches MBSE. Dabei unterstützt MBSE die Modellerstellung durch eine eigens hierfür entworfene Definitionsumgebung, die Systems Modeling Language SysML (siehe Kapitel 2.5.2).

Die Entstehung des modellbasierten Ansatzes ergab sich zwingend aus dem Bedarf an einer besseren Kommunikation zwischen diversen IT-Werkzeugen, die in der Produktentwicklung eingesetzt werden. Insbesondere mussten Medienbrüche zwischen den verschiedenen disziplinspezifischen Modellen beseitigt werden. Aus diesem Grund soll das Systemmodell schon bei der Aufnahme und Erfassung von Anforderungen angelegt und weiterhin stets unter Berücksichtigung aller disziplinspezifischen Entwicklungstätigkeiten erweitert und validiert werden. Erst dann kann das Systemmodell als sinnvolle Referenz für die domänenspezifischen Entwicklungsaufgaben herangezogen werden.

Durch die Verwendung des Systemmodells haben die Entwickler einen Zugriff auf stets aktuelle Informationen zum Projekt. Diese Informationen sind zentral verfügbar, zusammenhängend und konsistent. Außerdem sind diese Systeminformationen dank der Formalität eindeutig beschrieben und somit frei von den Interpretationsfehlern (Eigner, et al., 2014).

### 2.5.2 System Modeling Language (SysML)

Für das Systems Engineering wurde mit SysML eine Sprache entwickelt, die methodenunabhängig ist. Die Abkürzung SysML steht für Systems Modeling Language. Das International Council on Systems Engineering (INCOSE) hat sich im Jahr 2001 zum Ziel gesetzt, eine Standardsprache des Systems Engineering zu entwickeln (Weilkiens, 2006). Die Basis bildete die UML (Unified Modeling Language), die bereits in den 90er Jahren als grafische Modellierungssprache Eingang in die objektorientierte Softwareentwicklung fand. 2006 wurde SysML anerkannt und im nachfolgenden Jahr von der OMG

(Object Management Group) offiziell veröffentlicht. Seitdem unterliegt der Standard stetiger Überarbeitung und Weiterentwicklung (Kaiser, 2013).

Die wichtigsten Erweiterungen und Änderungen in Bezug auf UML (Version 2.1.1) sind im Folgenden aufgeführt: Es fand eine Umbenennung der Klassen in Systembausteine, des Klassendiagramms in ein Blockdefinitionsdiagramm (bdd) und des UML- Kompositionsstrukturdiagramms in ein internes Blockdiagramm (ibd). Es wurden zwei neue Diagrammarten wie Anforderungsdiagramm (req) und Zusicherungsdiagramm (par) sowie zahlreiche Erweiterungen in den einzelnen Diagrammarten eingeführt. So bekam das Aktivitätsdiagramm neue Modellierungselemente zur Abbildung von kontinuierlichen Abläufen und das interne Blockdiagramm zur Modellierung von Informationsflüssen. (Weilkiens, 2006)

Insgesamt bietet SysML neun verschiedene Diagramme, um das System möglichst genau aus allen Perspektiven und Sichten zu erfassen und zu dokumentieren (siehe Abbildung 39).

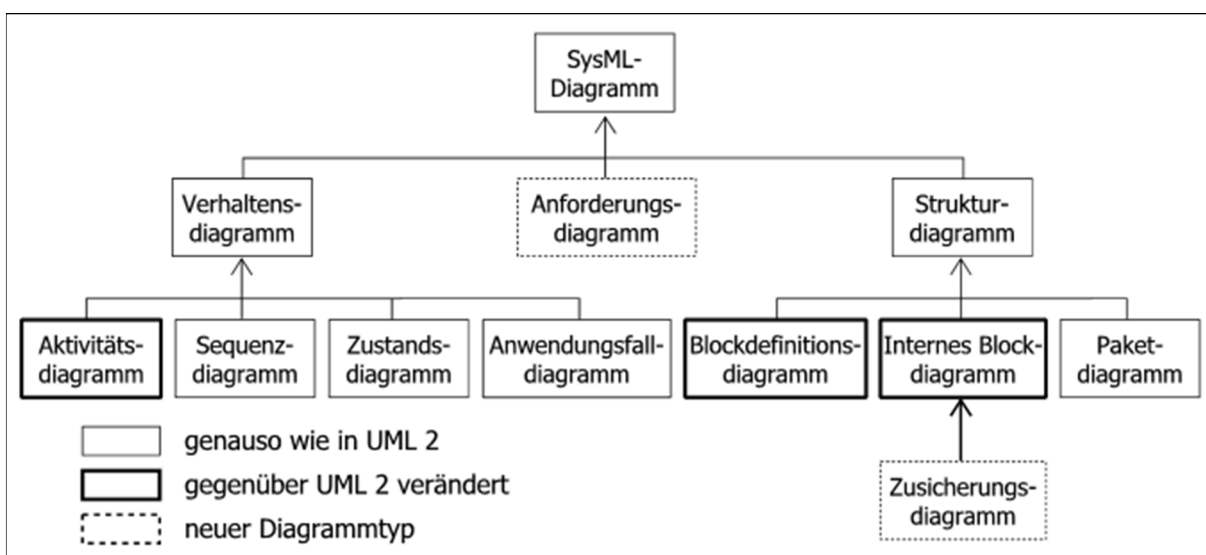


Abbildung 39: Diagrammarten in SysML (Partsch, 2010)

Systeme wie Kraftfahrzeuge oder selbst Einzelkomponenten wie Getriebe sind so komplex, dass ihre Modellierung in einem einzigen Produktmodell wenig hilfreich für die Erkennung der Zusammenhänge ist. Aus diesem Grund wird versucht, mit verschiedenen Sichten bestimmte Aspekte eines Systems gezielt zu betonen und andere absichtlich auszublenden. Die verschiedenen Diagramme in SysML entsprechen unterschiedlichen Sichten (z. B. strukturelle, funktionale, etc.) auf das System. Die Summe aller Sichten ergibt schließlich das Systemmodell (Rumpe, 2011). Die Vielfalt der diagrammatischen Sprachen soll eigentlich zur Komplexitätsbewältigung beitragen, kann aber auch erhebliche Nachteile aufweisen, denn verschiedene Sichten haben oft auch verschiedene Notationselemente. Damit steigen der Modellierungsaufwand sowie die Pflege der Modelle. Außerdem

wird dadurch die Konsistenzprüfung des Modells bei den Änderungen in einer Sicht, welche meistens einen Einfluss auf die Sachverhalte in den anderen Sichten hat, erschwert (Hitz, et al., 2005).

Für die im Rahmen der vorliegenden Arbeit gestellten Aufgaben der Prozessbeschreibung, sind folgende Diagramme von besonderem Interesse: das Aktivitätsdiagramm (act), das Anforderungsdiagramm und zwei Strukturdiagramme (bdd und ibd).

Das Aktivitätsdiagramm wurde fast vollständig aus UML übernommen. Die wichtigste am Aktivitätsdiagramm vorgenommene Erweiterung ist, dass die Modellierung kontinuierlicher Abläufe möglich geworden ist. Mit anderen Worten, die Beschreibung der kontinuierlichen Aspekte im System kann als «continuous» erfolgen. Für strömende Flüsse gibt es jetzt das Bezeichnungsattribut «stream». Zyklische Eigenschaften der diskreten sowie kontinuierlichen Flüsse können über «rate» angegeben werden. In Abbildung 40 sind einige dieser Erweiterungen zur Veranschaulichung dargestellt (Partsch, 2010).

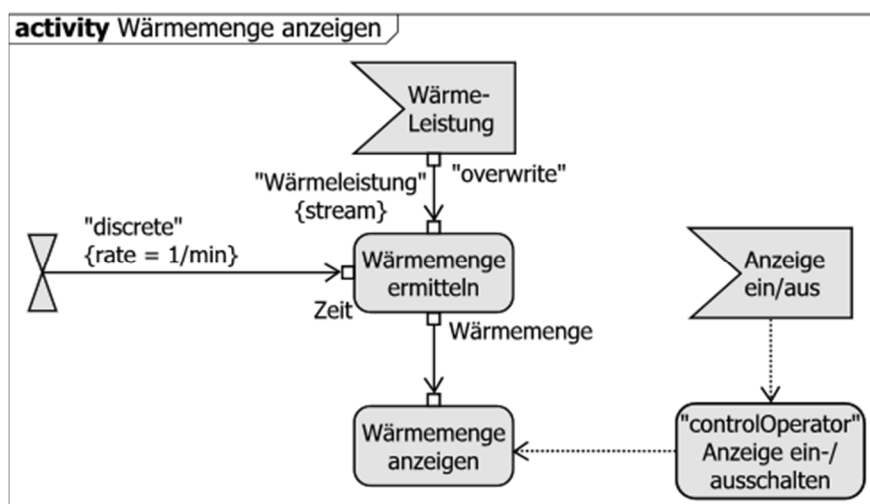


Abbildung 40: Beispiel mit Erweiterung des Aktivitätsdiagrammes (vgl. ebd.)

Anforderungen spielen im Systems Engineering eine Schlüsselrolle, denn sie bilden das Fundament für das gesamte Entwicklungsprojekt. Um diese zu modellieren, wurde in SysML eine neue Art des Diagramms eingeführt, das Anforderungsdiagramm. Anforderungen können dort als Anforderungselement <<requirement>> mit den Eigenschaften „ID“ und „TEXT“ beschrieben werden. Als Beispiel sind in der Abbildung 41 zwei Anforderungen „Dateneingabe“ und „Kundendateneingabe“ graphisch dargestellt. Die Anforderungen können auch in einer Tabellenform (siehe

Abbildung 42) dargestellt werden, was oft nützlich ist, da die Anzahl der Anforderungen oft sehr groß sein kann (Partsch, 2010).

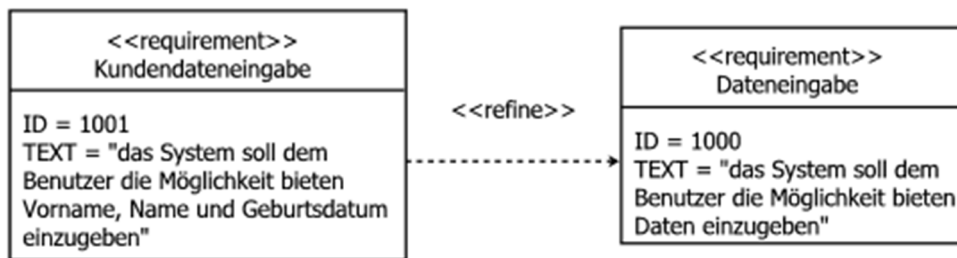


Abbildung 41: Beispiel eines Anforderungsdiagramms (JAXenter, 2008)

table Anforderungen in Tabellennotation		
ID	NAME	TEXT
1001	Dateneingabe	Das System soll dem Benutzer die Möglichkeit bieten Daten einzugeben.
1002	Kundendateneingabe	Das System soll dem Benutzer die Möglichkeit bieten Vorname, Name und Geburtsdatum einzugeben.

Abbildung 42: Anforderungstabelle zur Abbildung 41 (JAXenter, 2008)

Zur Definition der Beziehungen von Anforderungen können verschiedene Bezeichnungsattribute verwendet werden, zum Beispiel «satisfy», was „Anforderung erfüllen“ bedeutet. Dies wird beispielsweise für die Zuweisung der Anforderung zu einem Systembaustein verwendet (Partsch, 2010) (siehe Abbildung 43).

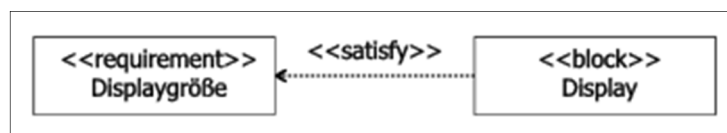


Abbildung 43: Beispiel der Zuweisung einer Anforderung einem Systembaustein

Wenn die Konkretisierung einer Anforderung durch eine andere Anforderung modelliert werden soll, wird das Attribut «refine» verwendet (siehe Abbildung 41). Außerdem stehen noch folgende Bezeichnungsattribute zur Verfügung (Kopytkov, 2009):

«deriveReq» - Anforderung ableiten, «trace» - Anforderung verfolgen, «verify» - Anforderung prüfen

Die Anforderungen selbst können in sieben verschiedene Typen unterteilt werden (OMG, 2015a):

«Requirement» – Basistyp, «Extended Requirement» – Verfeinerte Anforderung, «Functional Requirement» – Funktionale Anforderung, «Interface Requirement» – Schnittstellenanforderung, «Performance Requirement» – Qualitäts/Leistungsanforderung, «Design Requirement» – Anforderung ans Design, «Physical Requirement» – Anforderung an physikalische Merkmale.

Die Beziehungen können zwar in Tabellenform dargestellt werden, um aber die Beziehungen schneller nachvollziehen zu können, ist die diagrammatische Form besser geeignet.

Eine weitere Diagrammart ist das Strukturdiagramm, mit dessen Hilfe die Struktur des Systems beschrieben werden kann. Dies erfolgt über zwei Diagramme, das Blockdefinitionsdiagramm (engl.: block definition diagram - bdd) und das interne Blockdiagramm (engl.: internal block diagram - ibd). Systembausteine werden als Blöcke im Blockdefinitionsdiagramm definiert. Neben der Organisation können in dem Blockdefinitionsdiagramm auch die strukturellen Beziehungen wie zum Beispiel eine Assoziation zwischen den Blöcken modelliert werden. Mit Hilfe von Kardinalitätszahlen können dabei Informationen darüber angegeben werden, wie oft ein Element in einem anderen enthalten ist. Diese Angabe ist aber optional (Alt, 2012). Als Beispiel ist in der Abbildung 44 der Aufbau einer Solarstation mit Hilfe des Blockdefinitionsdiagramms dargestellt.

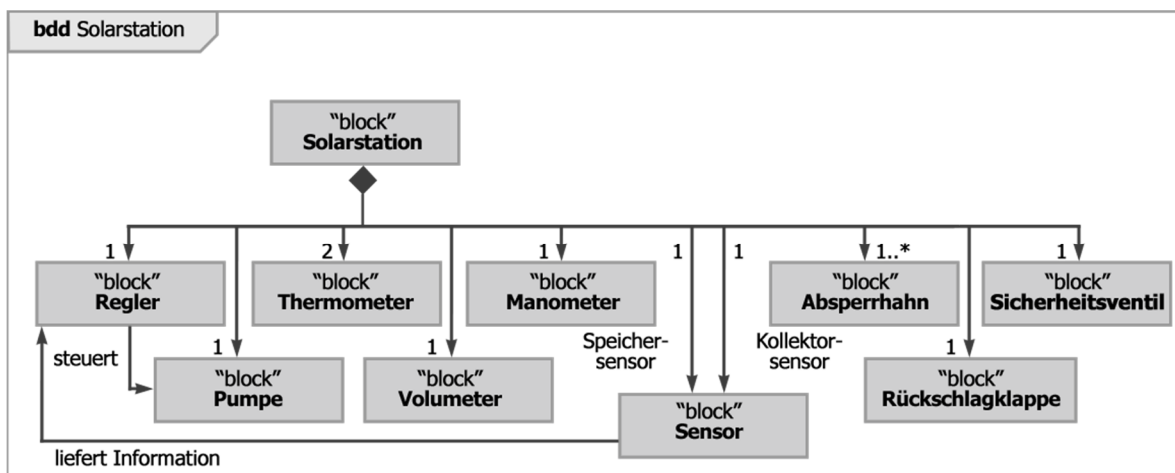


Abbildung 44: Beispiel eines Blockdefinitionsdiagramms (bdd) (Partsch, 2010)

Die Architektur eines Systems sowie die Kommunikationsbeziehungen zwischen den Blöcken sind in dem internen Blockdiagramm zu beschreiben. Diese Trennung ist auf den objektorientierten Ansatz zurückzuführen. Die Elemente in dem ibd verfügen über Schnittstellen (sogenannte Ports), welche der Interaktion mit den anderen Elementen dienen. Die Transferbeziehungen zwischen den Beteiligten im Diagramm werden mit Hilfe von Konnektoren angegeben, welche als gerichtete Kanten dargestellt werden. Im Allgemeinen ähnelt die graphische Notation sehr stark den entsprechenden UML Diagrammen. Ein Block wird in SysML genauso wie die Klasse in UML in Form eines weißen Rechtecks und ein Port als Quadrat dargestellt (Kaiser, 2013). In der Abbildung 45 ist die Darstellung des internen Blockdiagramms am Beispiel der Kommunikation der Komponenten der Solarstation zu sehen.

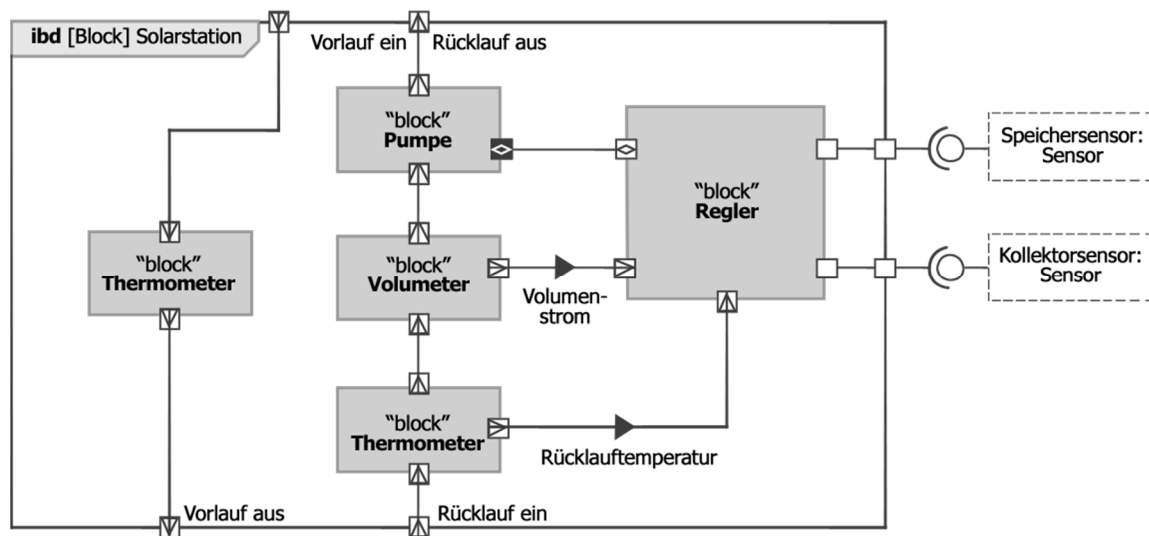


Abbildung 45: Beispiel eines internen Blockdiagramms (ibd) (Partsch, 2010)

Eine weitere Erweiterung gegenüber der UML-Sprache ist das Zusicherungsdiagramm (engl.: parametric diagram (par)). Mit Hilfe dieses Diagramms können verschiedene Eigenschaften von den Systembausteinen und ihrer parametrischen Beziehungen beschrieben werden. Parametrische Beziehungen werden in Form von mathematischen Gleichungen mit Parametern dargestellt. Das können physikalische Gesetze, einfache geometrische Formeln oder auch komplexe Übertragungsfunktionen sein. Die Definition der Gleichungen erfolgt mittels Zusicherungsbausteinen («constraint block»). Nach (Weilkiens, 2006) beschreibt eine Zusicherung in SysML eine Bedingung, welche vom System stets erfüllt sein soll. Damit ein Zusicherungsdiagramm erstellt werden kann, müssen zuerst die einzelnen Gleichungen als Zusicherungsbausteine modelliert werden. Als Beispiel ist in der Abbildung 46 der Zusicherungsbaustein „Pumpensteuerung“ dargestellt, welcher den Zustand der Pumpe beschreibt.

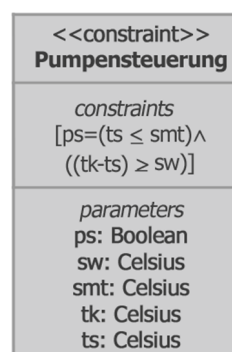


Abbildung 46: Zusicherungsbaustein „Pumpensteuerung“ (Partsch, 2010)

Dieser ist von der Speichertemperatur  $ts$ , der Kollektortemperatur  $tk$ , sowie von dem Schwellwert  $sw$  und der maximalen Speichertemperatur  $smt$  abhängig. Nach dem Definieren der Zusicherungsbausteine können die Verbindungen zwischen den entsprechenden Parametern der Zusicherungsbausteine und den Parametern der Systembausteine mit Hilfe von Bindungskonnektoren

modelliert werden. Die Angabe der Parameter von Systembausteinen kann mit Hilfe der Punktnotation erfolgen (siehe Abbildung 47)

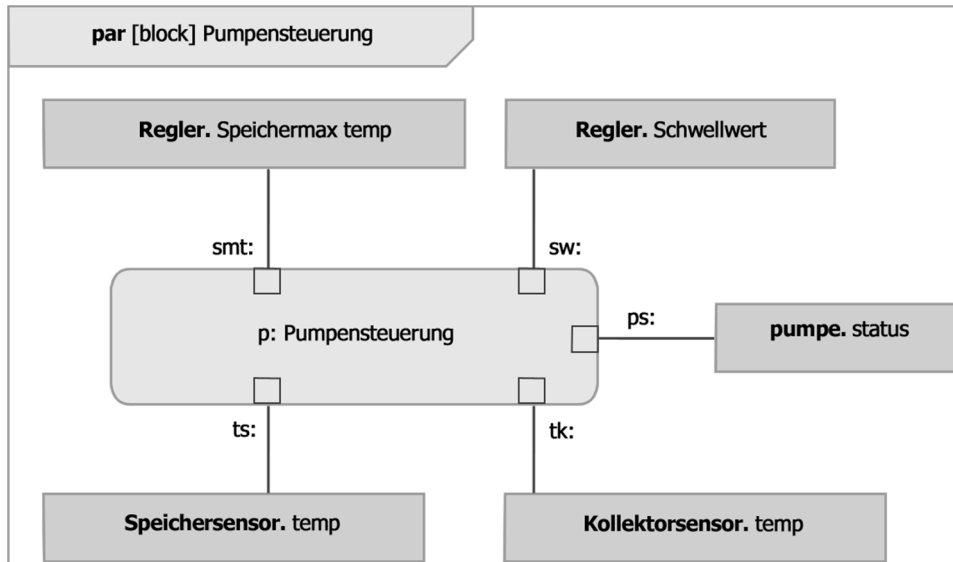


Abbildung 47: Beispiel eines Zusicherungsdiagramms (vgl. ebd.).

Eine der Aufgaben der Zusicherungsdiagramme ist, die mathematisch beschreibbaren Systemzusammenhänge zu erfassen und darzustellen. Außerdem kann das Ziel verfolgt werden, mit Hilfe der formal beschriebenen Systemzusammenhänge das Systemverhalten zu simulieren und somit die System- oder Komponentenauslegungen zu validieren. Leider ist die IT-Technische Unterstützung von solchen Zusicherungsdiagrammen in SysML noch nicht wirklich ausgereift, demzufolge wird aus heutiger Sicht von der Benutzung der parametrischen Zusicherungsdiagramme eher abgeraten und stattdessen empfohlen, dafür andere Werkzeuge aus dem Bereich der funktionsorientierten Entwicklung zu nehmen (Alt, 2012).

Ein weiteres Einsatzgebiet von Zusicherungen findet sich in dem Requirements Engineering. Die textbasierte Beschreibung der Anforderungen in SysML hat schon wegen der Ungenauigkeit der natürlichen Sprache bzw. der vielseitigen Interpretationsmöglichkeiten, einen klaren Nachteil. Andererseits lassen sich viele Anforderungen mit mathematischen Ausdrücken formalisieren und können dementsprechend mit den Zusicherungen abgebildet werden. Somit können Fehlinterpretationen der Anforderungen eliminiert werden, wodurch gleichzeitig die Basis für die spätere Verifikation geschaffen werden kann (oose., 2016). Dennoch kann die Methode des modellbasierten Systems Engineering (MBSE) als allgemeiner Leitfaden betrachtet werden, um komplexe Sachverhalte zu strukturieren und um diese so besser überschaubar, diskutierbar und lösbar zu machen. Das modellbasierte Systems Engineering hat mit SysML eine eigene Notation, stellt aber keine fachbezogene Vorgehensweise dar und ist damit allgemeingültig. Aufbauend auf dem Fundament des Anforderungsdiagramms können mit Hilfe von Aktivitätsdiagrammen die funktionalen Ebenen und mit Strukturdiagrammen die Verhaltensabhängigkeiten zwischen den Systemelementen

beschrieben werden. Damit ist diese Sprache prinzipiell geeignet, um die Berechnungsprozesse und die Beziehungen zwischen einzelnen Bauteilen und Baugruppen abzubilden. Aufgrund ihres Umfangs und ihrer Mächtigkeit benötigt die SysML-Sprache zu ihrer Anwendung vertiefte informationstechnische Kenntnisse und wird daher gegenwärtig nur eingeschränkt eingesetzt (Alt, 2012).

### 2.5.3 Einordnung der analytischen Berechnungen in MBSE

Mit dem System Engineering wird der Weg in Richtung eines einheitlichen und vollständig dokumentierten Entwicklungsprozesses eines Produktes beschritten. Der modellbasierte Ansatz MBSE erweitert diesen Weg um ein digitales Produktmodell, welches als Drehscheibe und Informationsfundament für alle Entwicklungstätigkeiten dient und die Entwickler immer mit den aktuellsten Informationen zum Entwicklungsprojekt aus allen beteiligten Disziplinen versorgt.

Die uneingeschränkte interdisziplinäre Kollaboration, insbesondere in den früheren Phasen des Entwicklungsprozesses, wird als Schlüssel zur Bewältigung der Komplexität bezeichnet, die in Zukunft in einem noch größeren Ausmaß auf die Entwicklung der technischen Produkte zukommt. Der MBSE Ansatz ermöglicht es schließlich, alle notwendigen Tests automatisch abzuleiten und auszuführen. Durch die Verwendung virtueller Prototypen kann der Entwicklungsprozess noch schneller und effektiver werden, indem viele der Entwicklungstätigkeiten, die heute an realen Objekten erfolgen, in die frühen Entwicklungsphasen der Produktentstehung verschoben werden (List, 2015).

Für viele Phasen des Entwicklungsprozesses nach dem Vorgehensmodell von Systems Engineering werden IT-Werkzeuge entwickelt, um den Austausch mit Daten aus externen Partnerschaften oder aus anderen Entwicklungsphasen und Domänen zu ermöglichen. Die dabei digital verfügbaren Informationen können zur Verifikation des Modells und seiner Komponenten sowie bei deren Integration genutzt werden. Bei der Entwicklung mechanischer Komponenten werden heute aber vorzugsweise die geometrischen Eigenschaften aus CAD-Modellen in Betracht gezogen. Als Beispiel ist in der Abbildung 48 die Implementierung des MBSE Ansatzes bei Daimler AG vorgestellt.



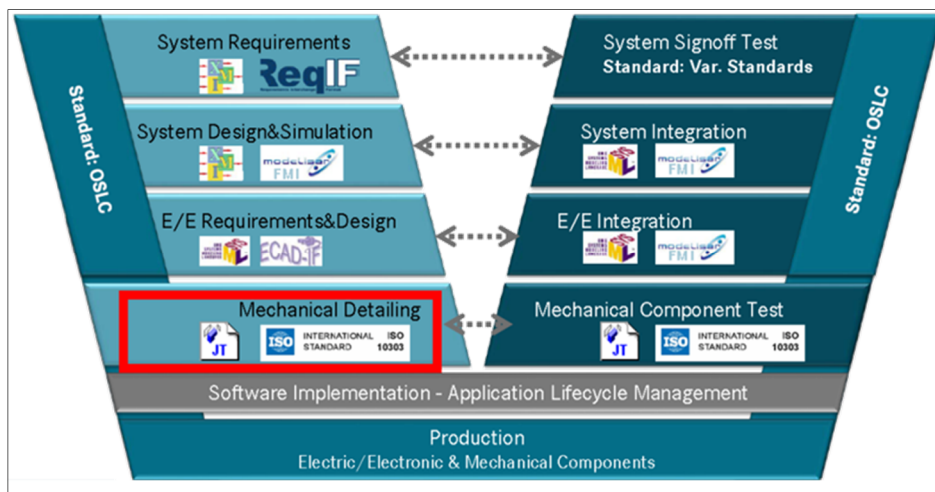


Abbildung 48: Systementwicklung nach V-Diagramm aus MBSE (Haasis, 2016)

Die mechanische Konstruktion ist in dem MBSE Ansatz lediglich mit der Geometrie und der Struktur im JT-Format und als Standard nach ISO 10303 angegeben. Da die Konstruktion eines Bauteils aber mit CAD-Systemen nicht komplett abgedeckt wird, sind viele analytische Verfahren notwendig, um den Entwurf zu realisieren, besonders bei Neukonstruktionen. Durch die Betonung der CAD-Tätigkeiten in der mechanischen Konstruktion geht die Fülle von Entscheidungen und analytischen Handrechnungen sowie den IT-unterstützten Auslegungen, die das Wissen der Ingenieure beherbergen, verloren. Der Bereich der analytischen Berechnungen ist dem CAD untergeordnet und wird als ein lokal vernetzter domänenspezifischer Prozess angesehen. Dieser fand bis jetzt keine Kopplung zum übergeordneten Produktentwicklungsprozess, welcher die produktrelevanten Informationen aus allen Phasen des Produktlebenszyklus von der Anforderungsaufnahme bis hin zum Recycling beinhaltet. Die fachliche Tiefe der Entwicklungsprozesse, die auf der Domäne spezifischen Ebene der mechanischen oder elektronischen Komponentenentwicklung herrscht, ist von den Systemarchitekten und Systemingenieuren nicht mehr zu begreifen. Diese müssen aber nahtlos in die Prozesse des Systemverhaltens und der Produktentwicklung eingebunden werden.

Die analytischen Berechnungen von Maschinenelementen erfordern eine formale Sprache, um in die Systembetrachtung (Simulation) nach dem Systems Engineering Ansatz einbezogen werden zu können. Wie die Sprachkonstrukte aussehen könnten, um diese Symbiose zu ermöglichen, gilt es noch genauer zu eruieren. Fest steht aber, dass die Berechnungsprozesse transparenter werden müssen, um die Analyse der möglichen Anbindung an die schon vorhandenen Produktentwicklungsabläufe durchführen zu können.

Praktische Versuche zeigen (Grätz und Rodig, 2016, Eigner und Schleidt, 2006), wie die Lücke zwischen den SysML Modellen und den physikalischen Modellen zumindest in Richtung der Simulation geschlossen werden kann. Ein veranschaulichendes Beispiel liefert die Zusammenarbeit der Technischen Universität Dresden, ESI ITI GmbH, IBM Deutschland und Knorr-Bremse Systeme für

Schienenfahrzeuge GmbH, München. Ziel des gemeinsamen Projektes ist es, das nach dem Systems Engineering Ansatz beschriebene Modell zu realisieren und damit auch die Anforderungen an das reale System direkt in die physikalische Simulation einfließen zu lassen (siehe Abbildung 49).

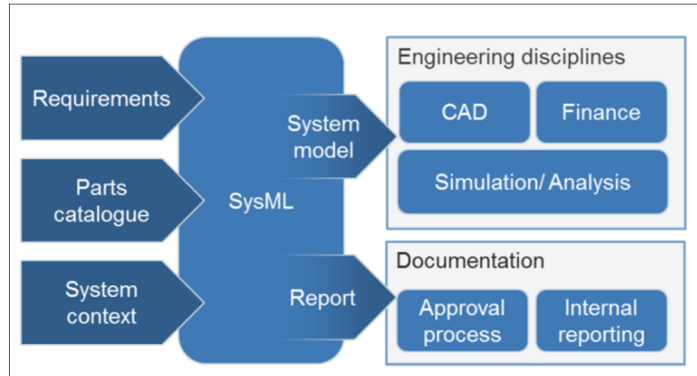


Abbildung 49: Integration der physikalischen Simulation in Systems Engineering (Grätz und Rodig, 2016)

Über spezielle Schnittstellen wird dabei versucht, die für das physikalische Modell benötigten Informationen aus den SysML Sichten zu entnehmen (Eigner, et al., 2011). Angefangen von den Anforderungen und Test Cases über die strukturelle Beschreibung bis hin zu der

Kommunikation zwischen den Komponenten werden alle relevanten Daten in die IT-Umgebung für die physikalische Simulation (in diesem Fall Modelica<sup>2)</sup>) transportiert.

#### 2.5.4 Modellbasierte vs. modellgetriebene Entwicklung

Wie aus dem Kapitel 2.5 entnommen werden kann, werden Modelle hauptsächlich zur Beherrschung der Komplexität herangezogen. Die verschiedenen Sichten des Modells erlauben die perspektivische Konzentration auf bestimmte Produkteigenschaften bzw. Problemstellungen, allerdings immer im Kontext der Systembetrachtung. Dieses Modell stellt gleichzeitig für alle Projektbeteiligten eine eindeutige und aktuelle Kommunikationsgrundlage und Informationsdrehscheibe zur Verfügung (siehe Abbildung 50). Erkennbar ist, dass das Modell nebenläufig existiert. Die Ausübung bestimmter Tätigkeiten unter der Nutzung des Systemmodells erfordert immer noch eine gewisse Interpretation vom Bearbeiter. Das Modell dient lediglich einer passiven Dokumentation. Die Änderungen am Systemmodell sind nicht sofort am Produkt ersichtlich, sondern müssen registriert und explizit am Produkt umgesetzt werden. Es entsteht eine Abstraktionslücke, wodurch die Beziehungen zwischen dem Modell und der Implementierung verloren gehen (Büchner, 2007). Diese Situation wird durch den Umstand verschärft, dass Software-Systeme eine immer höhere Komplexität aufweisen, die mit den bisherigen Mitteln nicht mehr beherrschbar ist. Zwar wurde der unübersichtliche Code mit Hilfe der diagrammatischen Sprachen in einer komprimierten und abstrakten Form abgebildet, um so die wichtigsten Aspekte schneller erfassbar zu machen, dennoch gibt es viele Routinetätigkeiten, die sehr fehleranfällig sind, da diese manuell erledigt werden müssen. Überdies ist der Aufbau des Codes sehr

2) Modelica – Modellierungssprache von Modelica Association, <https://www.modelica.org/>

oft durch individuelle Gewohnheiten und Fachkenntnisse geprägt, was eine lange Einarbeitung neuer Mitarbeiter erfordert (Abulawi, 2012, Kühne, 2007, Oestereich, 2005, Stahl, et al., 2007).

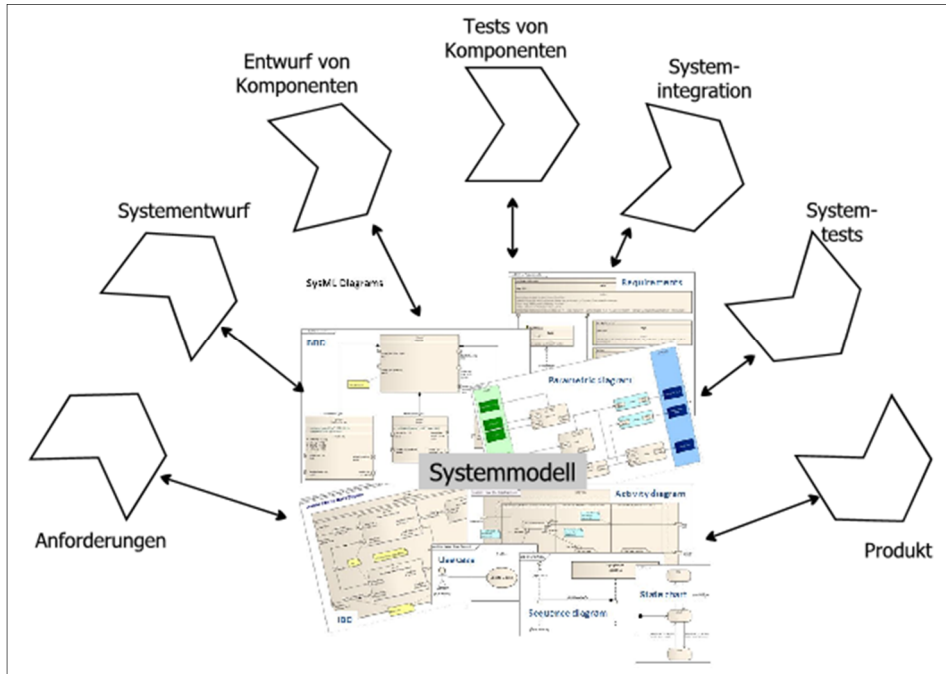


Abbildung 50: Modellbildung entlang des Produktentwicklungsprozesses

Liegt dagegen das Ziel in einer Automatisierung oder Teilautomatisierung der Konstruktionsabläufe, gerät die sogenannte „modellgetriebenen Entwicklung“ in den Fokus, die ihren Ursprung in der Softwareentwicklung hat und dort große Akzeptanz und Anwendung findet. Im Gegensatz zum modellbasierten Ansatz existiert bei dem modellgetriebenen Ansatz das Modell nicht mehr nebenläufig zur implementierten Softwarelösung, sondern bildet diese ganz oder in wesentlichen Teilen ab. Dies zwingt die Programmierer zu einer deutlich strukturierteren Arbeitsweise, bei der zuerst das Modell und erst dann der Code bearbeitet werden. Die Entwickler werden von den fehleranfälligen Routinetätigkeiten entlastet, da diese von der Entwicklungsumgebung übernommen werden. Auf Basis der im Modell formal beschriebenen Bausteine erfolgt automatisiert die Generierung des Codes, was zu einer weitgehenden Standardisierung und Vereinheitlichung des Codes führt (Stahl, et al., 2007). Die genannten Vorteile können auch auf die Konstruktionsabläufe übertragen werden (siehe Abbildung 51)

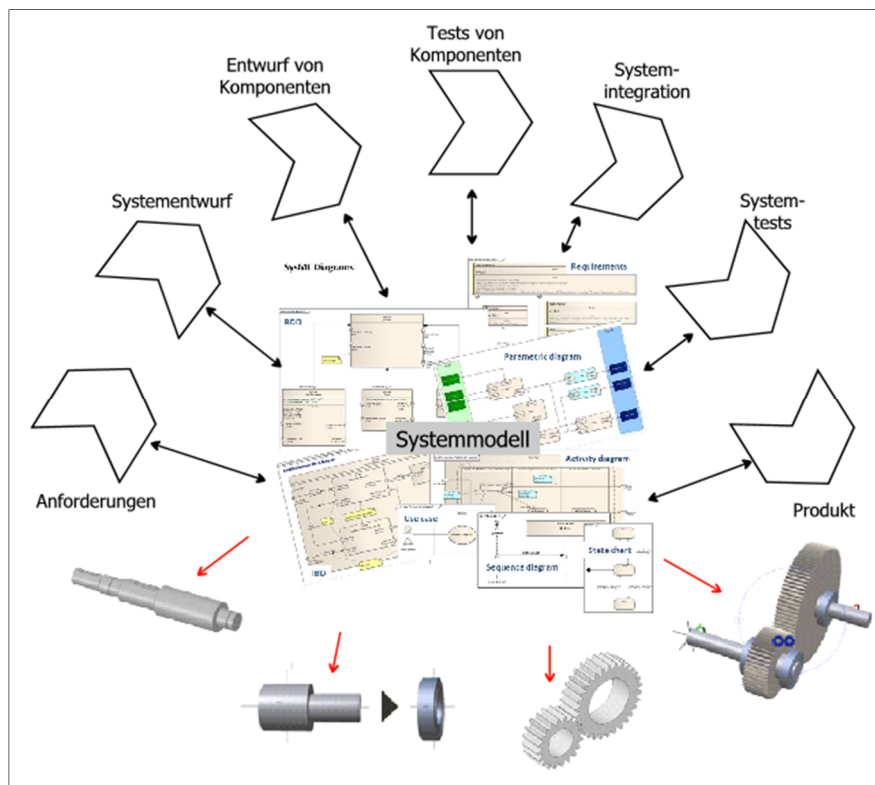


Abbildung 51: Modellgetriebene Konstruktion, adaptiert nach (Kühne, 2007)

Denn für viele Fragenstellungen existieren bereits bewährte Lösungen aus der Normung oder aus unternehmensinternen Vorschriften, in denen das technische Wissen im Hinblick auf den Lösungsprozess systematisch aufbereitet wurde. Mit Hilfe des formalen Modells kann dieses Wissen automatisch vom Rechner abgearbeitet werden, in dem nach vordefinierten Workflows einzelne Produktkomponenten automatisiert entworfen, simuliert und verifiziert werden. Ein gutes Beispiel dafür ist ein Getriebe, welches aus vielen verschiedenen Komponenten besteht.

## 2.6 Getriebeentwicklungsprozesses

### 2.6.1 Berechnung von Zahnradgetrieben

Ein Getriebe stellt eine komplexe und für verschiedene Maschinenelemente (Zahnräder, Wellen, Lager, Dichtringe, etc.) reichhaltige Baugruppe dar, deren Komponenten mehr oder weniger großen Einfluss aufeinander ausüben. Zu den grundlegenden Schritten des Entwicklungsprozesses eines Getriebes gehören zunächst, aufbauend auf den Anforderungen und getroffenen Annahmen, die Festlegungen des groben Getriebekonzeptes inklusive der Definition der einzelnen Getriebekomponenten. Danach folgt der erste Entwurf des gesamten Getriebes mit der Festlegung der Geometrie und Werkstoffeigenschaften aller Baugruppen und Baugruppenkomponenten. Die anschließende Nachrechnung soll die ermittelten Eigenschaften des Getriebes gegen die zu Beginn gestellten Anforderungen verifizieren und gegebenenfalls präzisieren. Zum Schluss werden IT-gestützte Simulationen (zum Beispiel FEM, MKS) und die am Rechenmodell nicht durchführbaren Analysen als

Messungen am realen Objekt vorgenommen, um das komplette Getriebe zu validieren (Schlecht, et al., o. D., Niemann und Winter, 2005) (siehe Abbildung 52).

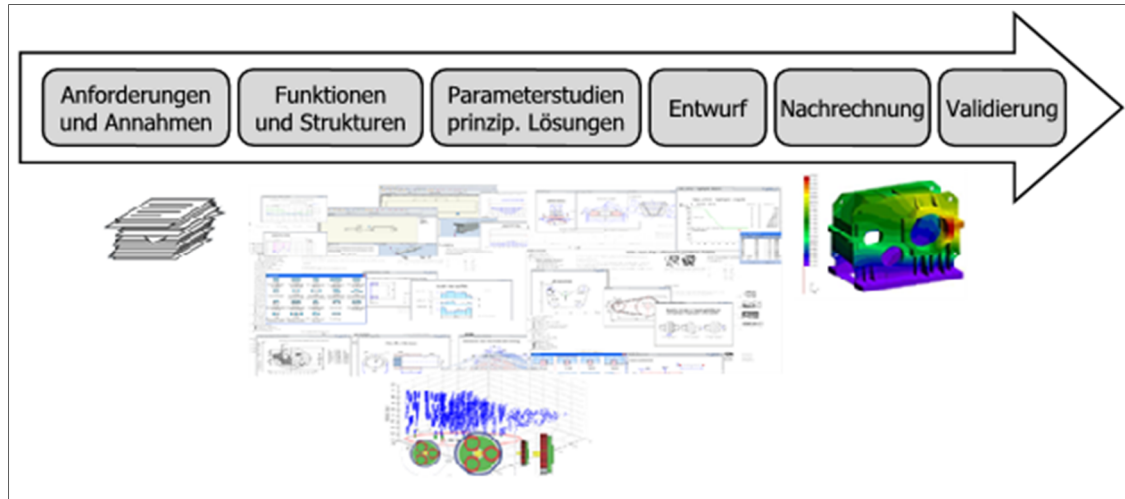


Abbildung 52: Klassischer Getriebe-Berechnungsprozess

Eine Getriebeentwicklung muss in Abhängigkeit von der Anwendung zahlreiche unterschiedliche Merkmale erfüllen, deren Ermittlung zum Teil hochspezialisiertes Wissen erfordert. Traditionell werden bei der Getriebeberechnung mehrere Berechnungsverfahren (Verzahnungsberechnung, Wellenberechnung, Lagerberechnung usw.) zunächst unabhängig voneinander durchgeführt (Polyakov und Gründer, 2013). Allein der Entwurf der Makro-Geometrie der Komponenten erfordert dabei oft schon mehrere Näherungsrechnungen. Die über eine Antriebswelle an einem Zahnrad angreifenden Kräfte werden an ein zweites Zahnrad, von diesem über eine Welle-Nabe-Verbindung an eine weitere Welle und anschließend auf ein oder mehrere Wälz- oder Gleitlager und schließlich auf das umgebende Gehäuse übertragen (siehe Abbildung 53). Da sich die Ergebnisse untereinander beeinflussen, müssen meistens mehrere Revisionen erfolgen. Die daraus resultierende Komplexität zu beherrschen und die Berechnungsergebnisse den geforderten Zielgrößen anzunähern, ist unter den gegebenen technologischen Möglichkeiten und normierenden Rahmenbedingungen eine große Herausforderung.

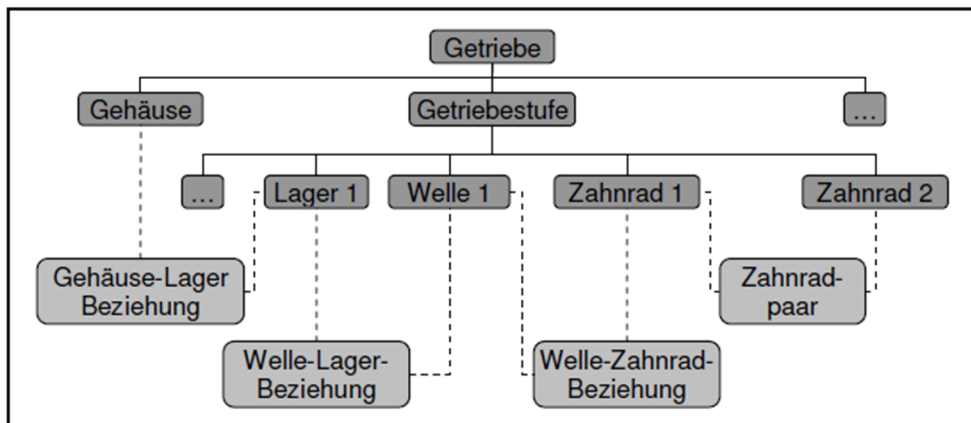


Abbildung 53: Getriebestruktur mit funktionalen Zusammenhängen (Bulligk, 2009)

Die erste und die wichtigste Phase im Getriebeentwicklungsprozess beinhaltet die Erfassung der Anforderungen, da diese Informationen das Fundament für den kompletten Entwicklungsprozess bilden. Wichtige Merkmale betreffen die technischen Aspekte wie Antriebsart, Lasten und Umgebungsbedingungen sowie die gewünschten Ausgangsparameter, wie Drehmoment, Drehzahl und Höhe des Geräuschpegels (Schlecht, et al., o.D.).

Als nächstes werden die funktionalen Gruppen ermittelt, die als Basis für die verschiedenen in Frage kommenden Prinzipzlösungen dienen. Hierbei muss entschieden werden, welche Getriebeart (Standgetriebe, Planetengetriebe, etc.) und welche Verzahnungsart bzw. Radform (Stirnradgetriebe, Kegelradgetriebe, etc.) gewählt werden soll. Außerdem muss entschieden werden, welcher Flankenlinienlauf (Gerad-, Schräg-, Doppelschräg- oder Bogenverzahnung) zum Einsatz kommen und wie viele Stufen das Getriebe haben soll. Weiterhin muss festgelegt werden, ob mehrere Getriebearten kombiniert werden sollen. Letztendlich sollte noch die Anordnung und die Lage der Getriebeelemente bestimmt werden. Mit diesen vorab getroffenen Überlegungen kann der mögliche Lösungsraum abgebildet werden. Je sicherer dieser analysiert und eingeengt wird, umso schneller kann eine optimale Lösung erreicht werden. Fehlentscheidungen zu Beginn der Auslegung können im weiteren Prozessverlauf nur sehr schwer wieder korrigiert werden. Maßgebende Annahmen wie zum Beispiel die Anzahl der Getriebestufen oder die Anordnung der Lager können später nicht mehr problemlos verändert werden, da die Auswahl aller weiteren Getriebeelemente hierauf aufbaut. Eine nachträgliche Veränderung kann zu einer fast kompletten Neukonzeption führen, welche mit entsprechendem Zeit- und Arbeitsaufwand verbunden ist (Jaroš, 2007, Bulligk, 2009).

Die Auslegung des Gesamtgetriebes untergliedert sich in die Auslegung der einzelnen Getriebekomponenten und stellt einen iterativen Prozess dar. Dieser beginnt bei den Anforderungen und wird in einem kontinuierlichen Optimierungsprozess auf Basis bewährter Nachrechnungsmethoden konkretisiert. Er endet schließlich mit der Erstausslegung eines Getriebes (siehe Abbildung 54).

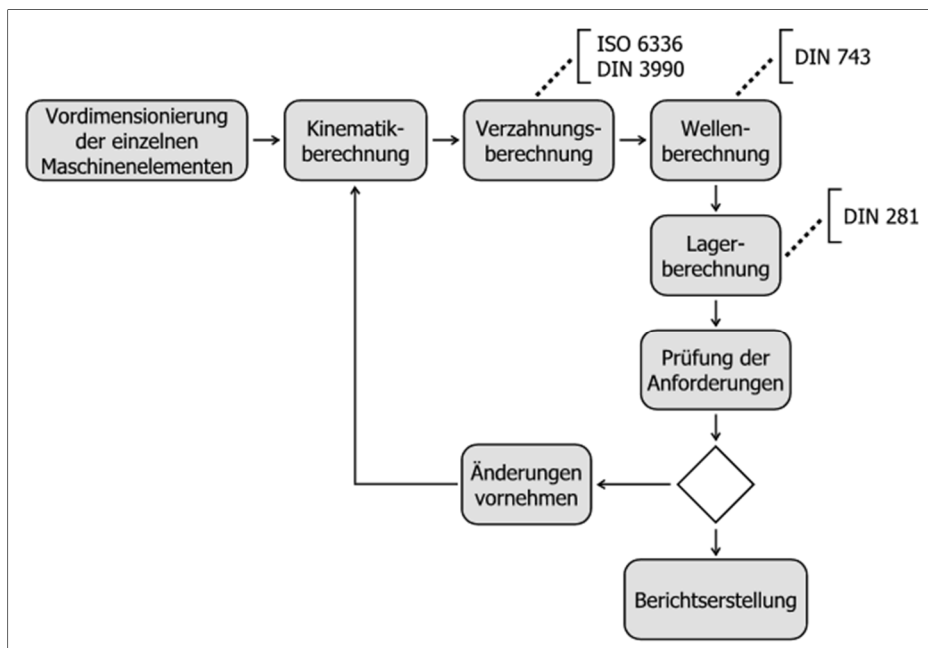


Abbildung 54: Schematischer Ablauf der Getriebeauslegung

Während für die Nachrechnung der Getriebekomponenten zahlreiche national sowie auch international anerkannte Rechenverfahren existieren (z. B. DIN, ISO, AGMA, usw.), wird bei der Erstausslegung der Getriebeelemente meistens auf firmeninterne Richtlinien oder Erfahrungen zurückgegriffen. Die Auslegungen sind grundsätzlich aufgabenspezifisch und dadurch sehr unterschiedlich. Aus diesem Grund beschränken sich die bekannten Normen auf die Nachrechnung und enthalten keine Auslegungsmethoden. Letztlich richten sich aber auch die Auslegungsmethoden an den Normen aus. Denn die erzielten Merkmale müssen letztendlich den anerkannten Regeln der Technik entsprechen. Die Vorgehensweisen bei der Auslegung von Maschinenelementen und Maschinenbau-Baugruppen sind im Allgemeinen durch iterative Verfahren und Parameterstudien geprägt.

Im Prinzip besteht die Auslegung des Getriebes aus einem iterativen Prozess, in dem Analyse und Synthese gleichermaßen vorkommen, wobei die Analyse hauptsächlich mit den entsprechenden Normberechnungen für die jeweiligen Getriebekomponenten erfolgt (Verzahnung mit ISO 6336 oder DIN 3990, Wellenberechnung mit DIN 743, etc.). Die Synthese basiert dagegen auf der Erfahrung und dem Wissen des Konstrukteurs. Nach jedem Analyseschritt muss er die Ergebnisse interpretieren können und gegebenenfalls die notwendigen Änderungen am Getriebemodell vornehmen. Dies ist keine einfache Aufgabe, denn die Fülle der zu berücksichtigenden Einflüsse infolge der starken funktionalen Vernetzung von Getriebekomponenten ist sehr groß. Die Erhöhung der Anzahl der Getriebekomponenten führt zu noch komplexeren Entscheidungsprozessen bei der Synthese. Eine Alternative bietet die Methode der expliziten Auslegung, welche mit Hilfe quantitativer Vorgaben der Parameter und formal beschriebener Zusammenhänge gezielte Berechnungen der gesuchten Größen



ermöglicht. Dies erfolgt durch die inverse Berechnung der im Analyseschritt durchzuführenden Normberechnungen (Parlow, et al., 2016) (siehe Abbildung 55).

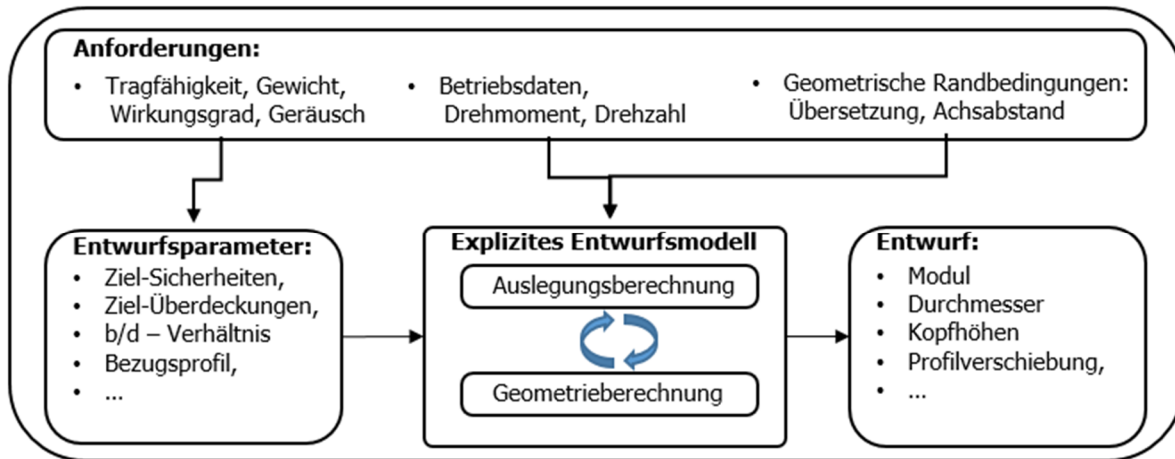


Abbildung 55: Explizites Getriebeauslegung (Parlow, et al., 2016)

Von entscheidender Bedeutung sind dabei die Startwerte (Entwurfparameter), die für die Bestimmung der komplexen und nichtlinearen Zusammenhänge zwischen den Parametern erforderlich sind (vgl. ebd.).

Die Verwendung dieser Methode sieht aber eine umfangreiche Vorarbeit vor, welche in der Aufstellung der mathematischen Zusammenhänge für die jeweiligen Suchgrößen liegt. Die Idee der inversen Berechnung der Zusammenhänge aus der Normung wird häufig zur automatisierten Auslegung von Getrieben und deren Komponenten verwendet, siehe (Pahl und Beitz, 1997, Parlow, et al., 2016, Jaroš, 2007, Bulligk, 2009).

## 2.6.2 EDV-Werkzeuge für Auslegungen und Parameterstudien

Parameterstudien für Einzelkomponenten, Unterbaugruppen oder Baugruppen wie z.B. der Berechnung einer Getriebestufe oder der Dimensionierung einer Welle-Nabe-Verbindung erfolgen gegenwärtig nur noch vereinzelt mit selbstgeschriebener Software. Überwiegend beruhen sie auf fertigen Zukauf-Applikationen, die entweder am Markt oder bei Forschungsvereinigungen verfügbar sind. Bezüglich der Eingabemöglichkeiten der zu berücksichtigenden Randbedingungen, der Genauigkeit, des Zusammenspiels verschiedener Elemente sowie der Reproduzierbarkeit der Ergebnisse sind jedoch aufgrund eigener Vorgehensweisen und Algorithmen des jeweiligen Anbieters Unterschiede zu verzeichnen. Außerdem sind diese Tools eher theoretisch orientiert und nicht immer in der Lage, solche Fragestellungen zu unterstützen, die eine gewisse Flexibilität der Randbedingungen erfordern. Dort, wo noch Tools eine Rolle spielen, die von eigenen Mitarbeitern mit Excel, Mathcad oder Programmiersprachen erstellt wurden, genügen sie oft nicht den Anforderungen an die Transparenz der Algorithmen und die Nachvollziehbarkeit der Ergebnisse. Auch bieten die



Benutzeroberflächen keine ausreichenden Erklärungskomponenten, denn diese IT-Werkzeuge wurden üblicherweise nur für die eigene Anwendung geschrieben und sind deshalb nur in seltenen Fällen vollständig dokumentiert, so dass ihre Benutzung durch Dritte sowie eine Pflege und Weiterentwicklung mit einem hohen Aufwand verbunden sind. Außerdem sind diese Tools für eine Integration in die Prozesskette einer Getriebeentwicklung meistens nicht geeignet.

Im Hinblick auf die Anforderungen des Systems Engineering erfordert insbesondere die Entwurfsphase eine flexible und für viele Aufgabenstellungen geeignete Berechnungsumgebung. Eine Bibliothek mit Standardelementen und ein Werkzeug zur schnellen Konfiguration von Berechnungsprozessen, die den Lösungsraum für die angestrebten Entwicklungsziele umreißen, die möglichen prinzipiellen Lösungen identifizieren und schließlich ein optimales Auslegungsergebnis präsentieren, würde dieser Aufgabe am ehesten gerecht. Zu berücksichtigen sind dabei Anforderungen der Usability, der Vernetzung mit der übrigen IT-Welt und der Entwicklungs-Community sowie integrierte Werkzeuge für die Dokumentation und Qualitätssicherung. Damit lassen sich die Ziele der nachfolgenden theoretischen Untersuchungen wie folgt definieren:

- Welche Informationen müssen die Grundelemente enthalten, aus denen sich die Berechnungsprozesse zusammensetzen?
- Wie ist der Aufbau der Grundelemente?
- Welche Beziehungen weisen diese Grundelemente untereinander auf?
- Wie lässt sich ein durchgängiger Prozess programmtechnisch zusammenbauen und
- wie die visuelle Sprache zur Prozessdefinition beschaffen sein muss?

### 3 Prozessgebundene Berechnungsbaugruppen

In den vorangegangenen Kapiteln wurden die Methoden und Prinzipien der Prozessbildung und deren Abbildung dargestellt und daraufhin untersucht, inwieweit sie die Automatisierung komplexer Berechnungen vorantreiben können. Die gewonnenen Erkenntnisse lassen darauf schließen, dass das Systems Engineering gemeinsam mit der bewährten Methode des Business Process Management die Berechnungsprozesse in der Konstruktion nachhaltig verbessern kann.

Daran anschließend soll mit der vorliegenden Arbeit eine Methode entwickelt und dargestellt werden, die sich an den Zielen der „Prozessgebundenen Berechnungsbaugruppen“ orientiert. Es handelt sich dabei um eine multidisziplinäre und systemübergreifende Betrachtung des beteiligten technischen Wissens im Hinblick auf:

- einen flexiblen Aufbau zusammengesetzter Berechnungsprozesse
- die Erhöhung der Transparenz der Entwicklung durch eine visuelle und systematische Handhabung
- sowie die Automatisierung der Abläufe und deren Dokumentation.

Die „smarte“ Dokumentation hält dabei Ergebnisse inklusive ihrer Entstehung fest.

Auf Basis der entwickelten Methode soll die Entwicklung eines rechnergestützten Systems vorgestellt werden. Mit dessen Hilfe können die Berechnungen verschiedener Maschinenelemente und damit zusammengestellte Baugruppen interaktiv schnell und effektiv durchgeführt werden.

Nachfolgend werden zunächst die theoretischen Untersuchungen dargestellt, welche die fachliche Basis der entwickelten Methode zur prozessorientierten Modellierung und Berechnung von Maschinenelementen und deren Baugruppen bilden. Ausgangspunkte sind die herkömmlichen, teilweise manuell und teilweise rechnergestützt ablaufenden Aktivitäten im Entstehungsprozess eines Getriebes. Sie werden unter Berücksichtigung der in Kapitel 2 vorgestellten Methoden und Vorgehensweisen unter strukturellen und ablaforientierten Gesichtspunkten neu gegliedert. Die Hauptidee der Realisierung der gesetzten Ziele der vorliegenden Arbeit besteht in der Ausarbeitung und Anwendung von formalisierten graphischen Modellen für die Organisation von Berechnungsprozessen der Maschinenelemente und der Dokumentation der Ergebnisse. Die zu implementierenden Berechnungsgrundlagen werden im Rahmen der Arbeit als bekannt angenommen. Die Hauptquellen liegen in der Normung, Literatur und Praxis.

Darauf aufbauend werden die allgemeinen Prinzipien und Eigenschaften des Berechnungsprozesses formuliert und dahingehend definiert und formalisiert, dass sie umfassend visualisiert werden können.

Aus den visualisierten Modellen werden schließlich rechnergestützt die ablauffähigen Berechnungsprozesse abgeleitet.

Ausgehend von den Zielen der vorliegenden Arbeit werden die Analyse und Anpassung der bekannten Techniken aus der Konstruktion an die zu entwickelnde Methode dargestellt. Dabei bilden die Konstruktionsaufgabe eines Produktes die Eingangsinformation und die Ausprägungen der Merkmale der Maschinenelemente und Baugruppen dieses Produktes die Ausgangsinformationen. Die internen Elemente der neuen Methode basieren auf den heute üblichen Konzepten der Organisation eines Berechnungsprozesses und sind schon im vorherigen Kapitel erläutert worden. Demzufolge wurden im Rahmen der Arbeit für die ausreichende Darstellung der Methode folgende Elemente zur Untersuchung und Formalisierung aufgenommen:

- Konstruktionsaufgabe und zu erfüllende Anforderungen,
- Beschreibung der Produkteigenschaften,
- Definition des funktionalen Berechnungselementes,
- Prozesse und ihre Aufteilung in Teilprozesse,
- Verzweigungen und Rückkopplungen in einem Prozess,
- Konsistenz der Einheiten in den Berechnungsmodellen,
- eine systematische Organisation von verschiedenen Berechnungsabläufen und
- Automatisierungselemente

Für die informationstechnische Realisierung der neuen Methode wurden SysML (Systems Engineering Markup Language) und BPMN (Business Process Management Notation) für die visuelle Darstellung von Prozessen sowie XML (Extensible Markup Language) im Hinblick auf das Datenmanagement und die formale Ablage eines Prozesses untersucht.

Im anschließenden praktischen Teil der Arbeit wird eine prototypische Realisierung des entwickelten Ansatzes am Beispiel des integralen EDV-Tools vorgestellt. Das wesentliche Augenmerk liegt dabei auf dem interaktiven Prozess-Designer, welcher die Organisation der Berechnungsprozesse ermöglicht. Am Beispiel einer Getrieberechnung wird anschließend die Funktionsweise veranschaulicht.

### 3.1 Theoretische Untersuchungen

Neben der Erhöhung der Transparenz der Berechnungsprozesse ist im Rahmen modellbasierter Konstruktionen deren automatisierte Ausführung unabdingbar. Nur dann kann eine höhere Akzeptanz durch die Ingenieure erreicht werden.

Bei dem modellgetriebenen Ansatz geht es darum, den Interpretationsspielraum zwischen dem Modell und dem Produkt zu eliminieren. Eine der Voraussetzungen dafür ist die Formalität der Modellierungsmethode. Aus diesem Grund werden nachfolgend alle Untersuchungsaspekte auch im Hinblick auf die Automatisierung beziehungsweise die formale Beschreibungsart analysiert.

#### 3.1.1 Anforderungen

Die Berechnungsprozesse einer Entwicklungs- bzw. Konstruktionsaufgabe können generell in zwei Arten untergliedert werden. Die erste beinhaltet die Auslegungsrechnungen, welche unter Berücksichtigung der ersten vorhandenen Daten aus der Anforderungserhebung die konzeptionellen Überlegungen (d. h. die ersten Lösungskonzepte) unterstützen. Weiter sind die Detailberechnungen bzw. die Nachrechnungen zu nennen, welche bei der Ausarbeitung und Detaillierung der einzelnen Komponenten sowie deren Integration in das Gesamtsystem durchgeführt werden.

Bereits zum Zeitpunkt der Ideendefinition eines Produktes werden Anforderungen an dessen Eigenschaften und Alleinstellungsmerkmale beschrieben. Die funktionalen Anforderungen sowie die Referenzen aus der Normung und dem Stand der Technik fließen in die Produktbeschreibung ein. Die Detaillierung der Anforderungen kommt meist erst mit dem ersten Lastenheft vom Kunden. Dieses beinhaltet die Dokumentation des Produktes aus der Kundensicht sowie die Beschreibung, welche Eigenschaften das zu entwickelnde Produkt aufweisen soll. Es werden die gestalterischen und Funktionskenngrößen sowie die allgemeinen Funktionseigenschaften angegeben. Sehr oft sind auch konkrete Lösungsprinzipien und Konzepte als gefordert vorgegeben. Dieses Vorgehen ist darauf zurückzuführen, dass heutzutage die meisten Entwicklungen durch Varianten-, Anpassungs- oder Wiederholkonstruktionen geprägt sind (Leemhuis, 2005).

Als nächstes wird auf Basis der Lastenheftinformationen, die das Soll-Produkt beschreiben, das Ist-Produkt spezifiziert. Dies erfolgt in einem Pflichtenheft, in dem der Hersteller aus seiner Sicht die Anforderungen an das Produkt in Bezug auf die Produktion und Qualität angibt. Nachdem in enger Zusammenarbeit mit dem Auftraggeber iterative Überarbeitungen erfolgt sind, wird das Pflichtenheft fertiggestellt und damit die Anforderungsliste für das Entwicklungsprojekt festgelegt. Das bedeutet aber noch nicht, dass die Anforderungsliste endgültig feststeht. Im Laufe der Entwicklung werden oft aufgabenabhängig weitere Anforderungen definiert. Diese können entweder die am Anfang zu allgemein gehaltenen Anforderungen konkretisieren oder so weit abändern, dass sich neue Konstellationen und Konstruktionslösungen ergeben (siehe Abbildung 56).

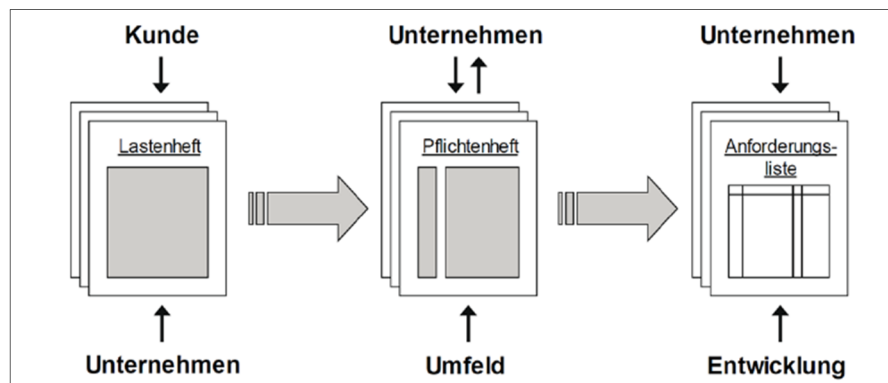


Abbildung 56: Entstehung der Anforderungen (Zingel, 2013)

Je besser eine Aufgabe beschrieben wird, umso höher ist die Wahrscheinlichkeit einer guten Lösung des Problems (Weilkiens, 2006). Aus diesem Grund ist es umso wichtiger, bei jedem Entwicklungsprozess die Kundenwünsche und technischen Forderungen sowie Randbedingungen (technologisch, umwelttechnisch, etc.) bereits zu Beginn der Entwicklung so genau und ausführlich wie möglich zu erfassen. Denn genau diese Aspekte werden als Teilziele und Randbedingungen der einzelnen Entwicklungstätigkeiten dienen.

Anforderungen unterscheiden sich aus Ingenieurssicht in erster Linie in technische und nichttechnische Anforderungen. Bei den nichttechnischen Anforderungen handelt es sich zum Beispiel um die Beachtung betriebswirtschaftlicher oder rechtlicher Aspekte. Sie beeinflussen oft nicht das Produkt selbst, sondern den Entwicklungs- und Produktionsprozess, darüber hinaus aber auch ökonomische und ökologische Bedingungen im späteren Betrieb. Die technischen Anforderungen beschreiben hingegen gezielt die Struktur oder Funktion des zu entwickelnden Produktes. Im weiteren Verlauf der hier vorliegenden Arbeit werden nur die technischen Anforderungen in Betracht gezogen.

Weiterhin können die Anforderungen in funktionale und nichtfunktionale Anforderungen unterteilt werden (siehe Abbildung 57).

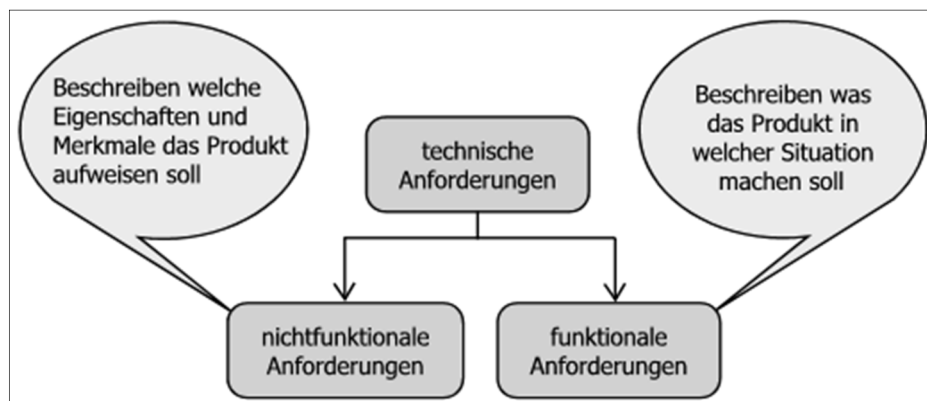


Abbildung 57: Technische Anforderungen

Die funktionalen Anforderungen rufen oft die nichtfunktionalen Anforderungen hervor. Als Beispiel für eine funktionale Anforderung soll folgendes Szenario genannt werden: „Bei einer Not-Situation soll sich die Maschine automatisch abschalten und einen Alarm auslösen“. Für eine nichtfunktionale Anforderung würde sich dann folgendes Wortlaut ergeben: „Die Maschine soll mit einer Sirene ausgestattet sein“. In der weiteren Konkretisierung würde folgende Anforderung aufkommen: „Die Lautstärke der Sirene der Alarmanlage muss mindestens 100 dB betragen“.

Im Weiteren soll die Darstellung der Anforderungen an die technischen Produkte aufgezeigt werden.

### 3.1.2 Darstellung von Anforderungen

Die bis heute meist textuell formulierten Anforderungen liegen zusammengefasst in der Anforderungsliste (siehe Tabelle 2)

ID	Name
A1	als Kraftmaschine soll ein E-Motor eingesetzt werden
A1.1	Nennleistung des Motors liegt bei 5 kW
A1.2	Nenndrehzahl liegt bei 3000 Umdrehungen pro Minute
A.1.3	Getriebe soll Vor- und Rückwärtstransport des Bandes ermöglichen
A1.4	Gesamtübersetzungsverhältnis soll sich maximal zweieinhalb Prozent von 12 unterscheiden
B2	Länge des Getriebes soll 800 mm nicht überschreiten
B2.1	Breite des Getriebes soll 600 mm nicht überschreiten
..	

Tabelle 2: Beispiel der Anforderungen an ein Getriebe

Diese technischen Anforderungen können schon an die ersten Lösungskonzepte gestellt werden. Mit Hilfe der modellbasierten Darstellung des Systems Engineering Ansatzes lassen sich diese Anforderungen zwar graphisch darstellen und zu den einzelnen Komponenten zuordnen (siehe Abbildung 58), jedoch ist eine automatisierte Ableitung der gewünschten Produkteigenschaften für ihre Weitergabe an die Konstruktionsabteilung damit noch nicht möglich.

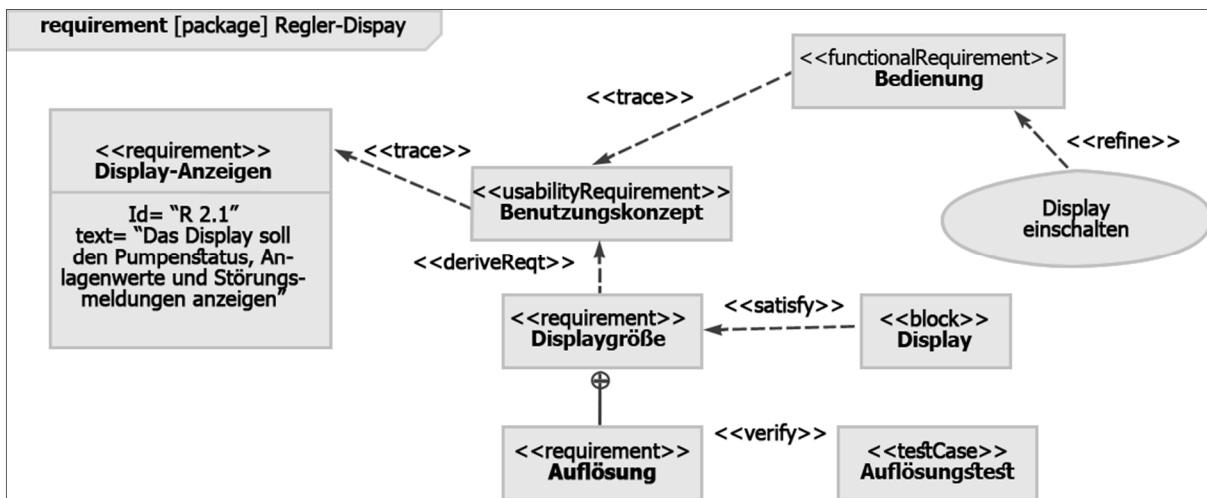


Abbildung 58: Beispiel eines Anforderungsdiagramms (Partsch, 2010)

Die Datenstrukturen des Anforderungsdiagramms in SysML lassen nur eine textuelle Beschreibung der Anforderung und die Identifikationsnummer (ID) zu. Für eine rechnergestützte Verarbeitung, wobei die Informationen ohne Zwischeninterpretation durch einen Menschen in den Konstruktionsprozess bzw. Berechnungsprozess einfließen können, müssen die Anforderungen dagegen formalisiert werden.

### 3.1.3 Formalisierung von Anforderungen

Die technischen Anforderungen beschreiben zunächst bestimmte Merkmale des zukünftigen Produktes, welche dann zu den geometrischen und physikalischen Eigenschaften dieses Produktes führen. Zusätzlich zur Benennung besitzen sie weitere Eigenschaften wie zum Beispiel einen Zahlenwert und eine Einheit.

Damit die Produkteigenschaften mit den Anforderungen beschrieben werden können, müssen die Eigenschaften von Anforderungen um einige Merkmale erweitert werden. Die Anforderung „maximale Länge des Getriebes soll 800 mm nicht überschreiten“ liefert dabei fast alle notwendigen Angaben (siehe Abbildung 59).



Abbildung 59: Produkteigenschaft aus Anforderung ableiten

Zusammengefasst ergibt sich folgende Anforderung (siehe Tabelle 3):

ID	Text	Benennung	Symbol	Zahlenwert	Einheit
A1	maximale Länge des Getriebes soll 800 mm nicht überschreiten	maximale Länge	Lmax	800	mm

Tabelle 3: Beispiel der Anforderungseigenschaften

Die Anforderung definiert in diesem Beispiel eine Randbedingung für das Merkmal „Länge“. Anstatt der direkten Ableitung der neuen Eigenschaft „maximale Länge“ kann an der Stelle die inhärente Eigenschaft „Länge“ des Objektes genommen werden. Dadurch wird die Anforderungsdefinition um weitere Zahlenwert-Eigenschaften erweitert (siehe Tabelle 4)

ID	Text	Objekt	Benennung	Symbol	Min. Zahlenwert	Zahlenwert	Max. Zahlenwert	Einheit
A1	maximale Länge des Getriebes soll 800 mm nicht überschreiten	Getriebe	Länge	L			800	mm

Tabelle 4: Beispiel der Anforderungseigenschaften erweitert

Diese Art der Formalisierung von Anforderungen ist nicht neu und wurde schon häufig empfohlen (siehe Düsing, 2001, Gausemeier, et al., 2014). In (Gausemeier, et al., 2014) wird sogar ein Verfahren vorgestellt, wie diese Formalisierung automatisch durchgeführt werden kann, in dem ein spezielles Programm die Struktur der Sätze in der Anforderungsbeschreibung erkennt, in die Satzfragmente zerlegt und mit Hilfe einer bestimmten Datenbasis in die formale Beschreibung umwandelt. Die vorliegende Arbeit beschränkt sich auf die manuelle Definition der Anforderungen mit jeweils vordefinierten Datenstrukturen.

Als nächstes gilt es, die zutreffende Definition der Produkteigenschaften bzw. der Merkmale zu ermitteln. Eine formale Beschreibung der Anforderungen fordert nicht nur eine korrekte Syntax, sondern auch eine eindeutige Semantik. Die Zuordnung der Anforderungseigenschaften zu den Produkteigenschaften und ihren Parametern ist nur durch eindeutig definierte Regeln für die Auswahl der Produktmerkmale möglich. Aus diesem Grund soll im folgenden Kapitelunterpunkt näher auf die Produktmerkmale eingegangen werden.

### 3.1.4 Produktmerkmale

Im Rahmen dieser Arbeit haben Merkmale eine sehr wichtige Bedeutung, denn sie beschreiben die Zielgrößen und Parameter der Berechnungsprozesse sowie die Schnittstellen zwischen diesen.

Laut DIN EN ISO 9000:2015-11 beschreibt ein Merkmal eine kennzeichnende Eigenschaft. Die Merkmale können inhärent oder zugeordnet sein, sowie qualitativ oder quantitativ. Zu den inhärenten Merkmalen zählen zum Beispiel die geometrischen und physikalischen Eigenschaften eines Objektes,



wie die Länge, der Durchmesser oder die Dichte. Diese sind immer vorhanden und werden auch als „innewohnend“ bezeichnet. Ein zugeordnetes Merkmal ist zum Beispiel der Preis eines Artikels. Weiterhin können die Merkmale beispielsweise in physikalische (mechanische, elektrische, etc.), sensorische (Geruch, Geschmack, etc.) und funktionale (z. B. maximale Geschwindigkeit eines Fahrzeuges) angeführt werden (DIN EN ISO 9000:2015-11).

Für den Berechnungsprozess sind die physikalischen und funktionalen Merkmale von besonderem Interesse, da sie die Eigenschaften darstellen, mit denen operiert wird. Das Operieren bedeutet „in einander umrechnen und ermitteln“. Die Tätigkeit zur Ermittlung eines oder mehrerer Merkmale und ihrer Merkmalswerte wird in der (vgl. ebd.) als „Bestimmung“ bezeichnet.

Aufschluss darüber, welche Merkmale ein bestimmtes Produkt charakterisieren und welche Merkmale welchen Objekten (zum Beispiel die Anforderung zur Einhaltung des minimalen Serviceintervalls von min. 10000 h eines Lagers in einem Getriebe) zugeordnet werden sollen, kann ein entsprechendes Datenmodell liefern. Auf diese Weise kann eine konsistente Beschreibung des Produktes bzw. der Anforderungen an ein zukünftiges Produkt erfolgen. Dies hilft zudem, interne Informationen aus verschiedenen Lebensphasen eines Produktes auf einander abzustimmen (siehe Abbildung 60).

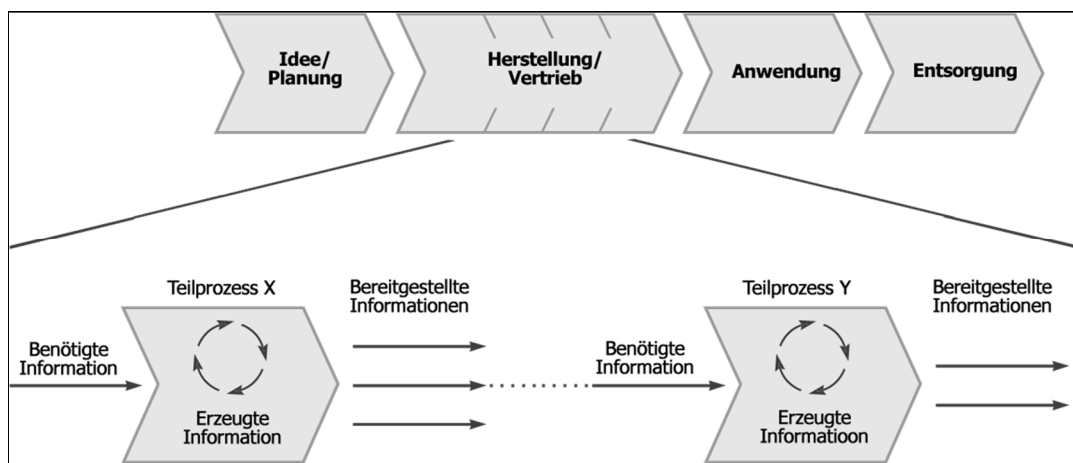


Abbildung 60: Lebenszyklus eines Produktes (DIN 4002-2:2007-07)

Ein Datenmodell ist demnach nicht nur für die fehlerfreie Weitergabe der Produktinformationen innerhalb eines Unternehmens notwendig, sondern auch, um nach außen hin bei der Kommunikation mit externen Partnern einen durchgängigen Informationsfluss zu garantieren (vgl. ebd.).

Aus der Vielfalt der verschiedenen Produkte und Erzeugnisse resultiert auch eine entsprechend große Vielfalt der domänenspezifischen Datenmodelle. Einige davon wurden bereits in der Normung aufgegriffen und zum Standard gemacht. So sind zum Beispiel Standards wie ISO 13584-1:2001-04 für die CAD-Teilebibliotheken und DIN EN 61360:2004-12 für die Beschreibung der elektronischen Bauteile entstanden. Die DIN 4002 Reihe hat auf Basis dieser zwei Normreihen versucht, eine

praktikable Lösung für den Aufbau eines Merkmalslexikons darzustellen. Die dabei erarbeiteten Attribute eines Merkmals werden in Hauptattribute (z. B. Benennung, Einheit, Quelle, Wert, etc.), Ergänzungsattribute (z. B. Symbol, Formel, Bildidentifikator, etc.) und Verwaltungsattribute (z. B. Kennung, Versionsnummer, Änderungsvermerk, etc.) unterschieden. Bei einigen Attributen, wie Benennung, Anmerkung, Quelle und Kommentar ist eine Übersetzung gefordert, um die Mehrsprachigkeit zu unterstützen (DIN 4002-2:2007-07).

Die Frage, die sich auch bei einem vorhandenen Datenmodell stellt, ist die nach der Definition der Produkteigenschaften. Außerdem ist von Interesse, wer die Vielfalt der möglichen Definitionen der Merkmale eines Produktes festlegen soll.

Im Bereich technischer Berechnungen wird der Datenfluss durch physikalische und geometrische Größen beherrscht. Kräfte werden (nach internationalem Einheiten System) in Newton und Drehmomente in Newton mal Meter angegeben. Durch die verteilten Arbeitsgruppen und die Vielfalt von Disziplinen, mit denen Aufgabenstellungen im Produktentwicklungsprozess konfrontiert werden, kann es allerdings dazu kommen, dass die Anforderungsbildung aufgrund unterschiedlicher Sprachen und Fachterminologien nicht eindeutig wird. Die Unterschiede könnten dazu führen, dass aus dem Moment ein Drehmoment wird und aus der Kraft [N] eine Antriebskraft [kN]. Dies würde automatisch zu falschen Randbedingungen der Produktentwicklung führen.

Zur Gewährleistung der Datenkonsistenz der Berechnungsprozesse müssen die Schnittstellen (Prozessvariablen) einheitlich und standardisiert sein. Sie sollen gleichzeitig flexibel und nach bestimmten Regeln ineinander überführbar sein, damit es nicht zu Einschränkungen bei der Modellierung kommt. Das kann mit einer Klassifizierung erreicht werden (siehe nächstes Kap. 3.1.5).

### 3.1.5 Klassifizierung von Produktmerkmalen

Generell können Klassifikationen zu unterschiedlichen Zwecken eingesetzt werden. Die Verwendung unternehmensweiter Klassifikationen ist beispielsweise eine Voraussetzung für die interne Verständigung. Besonders wichtig sind Klassifikationen aber auch für den externen Datenaustausch mit Partnern und Kunden, bei der Stammdatenpflege und in Bezug auf elektronische Kataloge. Stellt ein Zulieferer einem Unternehmen die Produktinformationen nach einer festen Standardklassifikation bereit, so kann das Unternehmen die so beschriebenen Produkte unmittelbar bei sich verwalten (Institut der deutschen Wirtschaft Köln, 2010).

Heutzutage existieren für die Klassifizierung von Produkten und Dienstleistungen viele Lösungen. Sie werden sowohl für den internen, wie für den externen Informationsaustausch genutzt.

Die bekanntesten davon werden im Folgenden aufgeführt und kurz erläutert:

- United Nations Standard Products and Services Code - UNSPSC
- Elektro Technisches Informations Modell - ETIM
- proficl@ss
- eCl@ss

UNSPSC wurde von den Vereinten Nationen sowie dem Unternehmen Dun & Bradstreet entwickelt und wird insbesondere im amerikanischen Raum eingesetzt. Dieses Klassifizierungssystem beschränkt sich auf die begriffliche Beschreibung von Erzeugnissen und Erzeugnisgruppen. Es verfügt auf der unteren Ebene über keine Sachmerkmale, mit denen Produkte näher beschrieben werden können und ist daher eher für das Warengruppenmanagement geeignet (ebd.).

ETIM – entstand als Initiative von ETIM e.v. Deutschland zur Standardisierung von elektronischen Gütern im Fachbereich Elektrotechnik. Der Vorteil des Systems liegt in der ausführlichen Merkmalsliste, die jeder Klasse zugeordnet ist. Diese macht die Beschreibung eines Produktes eindeutig. Die Klassifizierung ist auf die Elektrobranche ausgerichtet und demnach weniger für die Beschreibung der hier gestellten Aufgabe der Modellierung von Maschinenelemente-Berechnungsprozessen geeignet (vgl. ebd.).

proicl@ss ist eine branchenübergreifende und unabhängige Initiative des Vereines proficl@ss International e.V. und beschäftigt sich ebenfalls mit der Klassifizierung von Produkten. Das Hauptziel der Initiative ist ein branchenübergreifender elektronischer Handel mit eindeutig klassifizierten Produkten. Insgesamt beinhaltet proficl@ss elf verschiedene Teilklassifikationen, wobei die Beschreibung der elektrotechnischen Komponenten von ETIM übernommen wurde. Für die Maschinenelemente beinhaltet dieses System gegenwärtig noch keine ausreichenden Klassen und Eigenschaften, um sie in die Überlegungen im Rahmen der hier vorliegenden Arbeit mit einbeziehen zu können (Institut der deutschen Wirtschaft Köln, 2010, Uiterwyk, 2012).

eCl@ss – ist ein internationaler Standard für die Klassifizierung und Beschreibung von Produkten und Dienstleistungen. Auch hier ist das Ziel, die Vereinfachung des elektronischen Handels voranzutreiben. Entwickelt und gepflegt wird dieser Standard von dem gleichnamigen Verein (eCl@ss e. V.). Derzeit (2016) beinhaltet eCl@ss die größte Palette an klassifizierten technischen Produkten. Außerdem sind alle Begrifflichkeiten in mehreren Sprachen (insgesamt 14) vorhanden, so dass auch eine Landesgrenze überschreitende Produktplanung ermöglicht wird. Genauso, wie proficl@ss, ETIM und UNSPSC, gehört auch der eCl@ss Standard zu den hierarchisch aufgebauten Klassifizierungssystemen. eCl@ss besteht insgesamt aus 4 Ebenen, dem Sachgebiet, der Hauptgruppe, der Gruppe und der Untergruppe (siehe Abbildung 61) (eCl@ss e.V., o. D., Institut der deutschen Wirtschaft Köln, 2010).

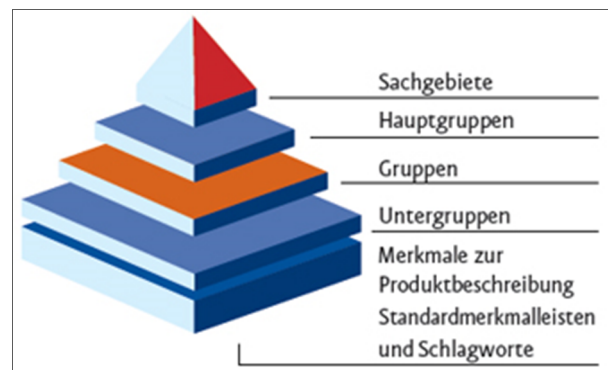


Abbildung 61: Aufbau eCl@ss (eCl@ss, 2016)

Die Klassifizierung beginnt von oben nach unten. Die übergeordnete Klasse entspricht immer der Vereinigung der untergeordneten Klassen. Auf der letzten Ebene (Untergruppen) bietet eCl@ss genauso wie proficl@ss klassenspezifischen Merkmalsleisten, mit deren Hilfe eine detailliertere Beschreibung der Produkte ermöglicht wird (Institut der deutschen Wirtschaft Köln, 2010, Maune, 2002). Zum Beispiel hat ein Stirnrad in der Vier-Ebenen-Hierarchie von eCl@ss eine Kennung bzw. Klassifikationsnummer „23-17-01-01“. Die Abbildung 62 gibt Aufschluss darüber, wie diese Kennung zusammengesetzt wird.

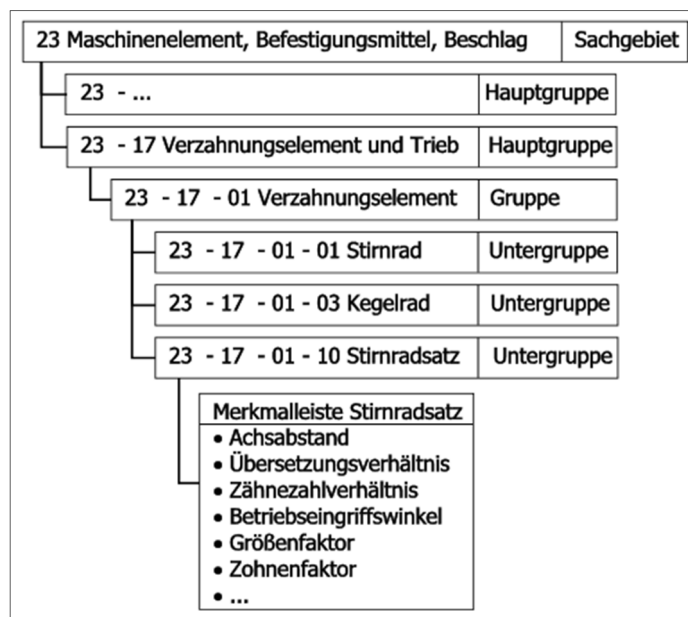


Abbildung 62: Beispiel der Klassifikation der Verzahnung nach eCl@ss-System

Die Gegenüberstellung der genannten Klassifizierungssysteme in Bezug auf das Einsatzgebiet sowie den Umfang der Klassen und deren Merkmale zeigt eindeutig die Vorteile des eCl@ss Standards. UNSPSC Standard deckt heute zwar die meisten Branchen ab, scheidet aber in diesem Vergleich schon durch die fehlenden Merkmalsleisten aus. Die große Sammlung der produktspezifischen Merkmale in diversen Branchen ist in eCl@ss nicht zuletzt der Harmonisierung mit ETIM, proficl@ss und PROLIST zu verdanken. Das auf DIN ISO 13584-42 und IEC 61360-2 basierende Datenmodell von eCl@ss sorgt

zudem nicht nur für die Investitionssicherheit, sondern auch für die einheitlichen Schnittstellen mit den anderen Systemen. Außerdem stellt der eCl@ss Standard die normgerechten Merkmale zur Verfügung. Durch die Kooperation von eCl@ss mit dem Deutschen Institut für Normung (DIN) soll sichergestellt werden, dass die eCl@ss Merkmale über den DIN-Server der nationalen und internationalen Normung zugeführt werden können (Institut der deutschen Wirtschaft Köln, 2010). Damit wurde der eCl@ss Standard als Basis für die im Rahmen der Arbeit gestellten Klassifikationsaufgabe ausgewählt.

Bezüglich der Klassifizierung der technischen Produkte wurden im eCl@ss-System jedoch einige Unstimmigkeiten eruiert. Zum Beispiel wird die „Übersetzung“ (02-BAB640) in eCl@ss als Merkmal der Klasse Stirnrad geführt. Analog gilt auch der „Achsabstand des Radpaares“ (02-BAB639) als Merkmal des Stirnrades, obwohl diese beiden Eigenschaften der in der eCl@ss vorhandenen Klasse Stirnradsatz bzw. Planetengetrieberadsatz angehören sollten. Diese Unstimmigkeit kann auf die Tatsache zurückgeführt werden, dass die Klassifizierung zunächst für kaufmännische Zwecke entwickelt wurde und demzufolge auch genutzt wird, so dass die eigentliche Klassifizierung nicht immer durch die Fachingenieure erfolgt. Aus diesem Grund werden die Merkmale in der vorliegenden Arbeit nicht vollständig aus eCl@ss übernommen, sondern in Anlehnung an eCl@ss.

### 3.1.6 Merkmalseigenschaften eines Objektes

Die Formalisierung der Anforderungen liefert den ersten Teil der minimal notwendigen Informationen über die Eigenschaften von Merkmalen eines bestimmten Objektes, wie zum Beispiel das zu entwickelnde Produkt oder seine Komponenten. Der zweite Teil beinhaltet die Klassifizierung der Eigenschaften dieses Objektes. Den abschließenden Teil liefert die graphische Repräsentation der Anforderungen. Insgesamt werden abgesehen von den Eigenschaften, welche für die Darstellung des Wertes auf der GUI (Graphical User Interface) - Seite notwendig sind, die folgenden Merkmalseigenschaften festgehalten:

- Klassenidentifikator (ID der Klasse, der das Merkmal zugeordnet ist)
- Kennung (eineindeutige sechstellige Identifikationsnummer)
- Definition (Kurzbeschreibung mit max. 1000 Zeichen)
- Benennung (orientiert, an den in den Normen gebräuchlichen Benennungen)
- Kurzbezeichnung (gekürzte, aber trotzdem sinnvolle Darstellung der bevorzugten Benennung mit möglichst wenig Zeichen, welche mit dem Symbol übereinstimmen darf)
- Bevorzugtes Symbol (alphanumerisches Zeichen, Bildzeichen oder Kombinationen nach internationalen Normen. Das Bevorzugte Symbol muss einem in der Branche üblichen Formelzeichen entsprechen. Existieren Normen für diese Symbole, sind diese anzuwenden)
- Ersatzsymbol (Die Auflistung eines Ersatzsymbols ist mitunter aus historischen Gründen zweckmäßig)

- Abbildung (Die graphische Darstellung verdeutlicht die Bedeutung eines Elements bzw. ergänzt eine schriftliche Definition in Form einer illustrierten Festlegung)
- Formel (Mathematischer Zusammenhang mit anderen physikalischen Größen, entsprechend einer Rechenvorschrift zur Darstellung der Semantik)
- Quelle der Definition (Verweis auf ein öffentlich zugängliches (zitierfähiges) Ursprungsdokument, aus dem die Definition für das Merkmal ganz bzw. sinngemäß übernommen wurde)
- Basis-Maßeinheit (SI-Einheiten sind bevorzugt zu verwenden. Multiplikatoren bei den Einheiten (z. B. Mega oder Milli) dürfen immer dann merkmal-spezifisch verwendet werden, wenn entsprechendes von normativer Seite festgelegt ist.)
- Maßeinheit (zur Darstellung auf dem graphischen Interface verwendete Einheit)
- Datentyp (nach Tabelle 5)
- Wert / Min., Wert / Max., Wert (In Zahlen oder Zeichen ausgedrücktes Datenelement, das als Ausprägung für ein Merkmal übertragen wird.)

Wert	Bedeutung
Integer	Eine ganze Zahl, die als Dezimalzahl als eine Abfolge von Ziffern (Symbole 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9) mit einem Vorzeichen darstellbar ist.
Real	Eine rationale und irrationale Zahl, die als Dezimalzahl als eine Abfolge von Ziffern (Symbole 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9) mit einem Trennzeichen (Dezimalpunkt oder Komma) zwischen Vorkomma- und Nachkommastelle(n) darstellbar ist.
String	freier Fließtext mit allen zulässigen Satzzeichen ohne Steuerzeichen
Übersetzbarer String	Freier Fließtext mit allen zulässigen Satzzeichen ohne Steuerzeichen. Bei Werten vom Typ „Übersetzbarer String“ wird zur Interpretation zusätzlich ein Sprachenindikator benötigt

Tabelle 5: Mögliche Datentypen eines Merkmales (vgl.: DIN 4002-3:2006-3)

Ein sehr wichtiger merkmalsbezogener Begriff ist die „Ausprägung eines Merkmals“, welcher einen konkreten Wert eines Merkmals beschreibt. Unterschieden wird dabei zwischen den quantitativen Merkmalen, die einen Zahlenwert inkl. der Einheit als Wert haben und den nicht-quantitativen Merkmalen, die als Wert einen Freitextangabe bzw. einen Eintrag aus der Zuordnungstabelle haben (DIN 4002-2:2007-07). Zur Sprachregelung wird im weiteren Verlauf der Arbeit der Begriff „Wert“ des Merkmals (unabhängig, ob quantitativ oder nicht-quantitativ) anstatt Ausprägung des Merkmals benutzt.

### 3.1.7 Einheitensystem von Merkmalen

Ein wichtiger Punkt der Beschreibung von Merkmalen ist ihre Einheit, da diese bei der Berechnung von physikalischen Größen eine entscheidende Rolle zur Beurteilung der Ergebnisse spielen kann. Außerdem ist dieser Aspekt von großer Bedeutung für die spätere Implementierung des Prozesssimulators. Abhängig von der Aufgabenstellung bzw. deren Randbedingungen (zum Beispiel Objektgröße) ist es bei einigen Zahlenwerten viel komfortabler eine bestimmte Einheit zu verwenden, um den Zahlenwert visuell schnell erfassen zu können. So wirken zum Beispiel in einem Windturbinengetriebe Verzahnungskräfte im kN Bereich. Im Gegensatz dazu sind diese bei einem Uhrwerk einer Armbanduhr äußerst klein und bewegen sich im  $\mu\text{N}$ -Bereich. Die Berechnungsfunktionen in den Funktionsbausteinen werden aber so definiert, dass die Berechnung nur mit bestimmten Einheiten plausibel ist. Ein weiteres Beispiel ist die Angabe des Winkels in Grad. Die Größe des Winkels von  $30^\circ$  ist jedem sofort verständlich, im Gegensatz zu 0.523598 Radiant. Für die Beschreibung der mathematischen Zusammenhänge in einer Programmiersprache ist aber genau diese Radiant-Einheit wichtig.

Für die Angabe der Einheiten nach SI-Einheitensystem existieren insgesamt sieben Basis-Einheiten und 22 davon abgeleitete Größen (siehe Tabelle 6 und Tabelle 7). Im Rahmen der vorliegenden Arbeit werden die Basisgrößen und die davon abgeleiteten Größen zu den Transfergrößen zusammengefasst.

Größe	Einheit
Länge	m
Masse	kg
Zeit	s
Elektrische Stromstärke	A
Temperatur	K
Stoffmenge	mol
Lichtstärke	cd

Tabelle 6: Basisgrößen

Größe	Einheit	In SI-Basiseinheiten ausgedrückt
Ebener Winkel	rad	1
Raumwinkel	sr	1
Frequenz	Hz	$\text{s}^{-1}$
Kraft	N	$\text{m} \cdot \text{kg} \cdot \text{s}^{-2}$
Druck	Pa	$\text{m}^{-1} \cdot \text{kg} \cdot \text{s}^{-2}$
Energie, Arbeit, Wärmemenge	J	$\text{m}^2 \cdot \text{kg} \cdot \text{s}^{-2}$
Leistung	W	$\text{m}^2 \cdot \text{kg} \cdot \text{s}^{-3}$
...	...	...

Tabelle 7: Abgeleitete Größen (DIN 1301-1:2010-10)

Da die unterschiedlichen Parameter mit identischer Basisgröße (z. B. Länge) in Abhängigkeit von der jeweiligen Aufgabenstellung in unterschiedlichen Größenordnungen (mm oder m) angegeben werden können, müssen diese Größen, um die Verknüpfung zwischen den Funktionsbausteinen konsistent zu halten, in eine kohärente Einheit umgerechnet werden (siehe Abbildung 63).

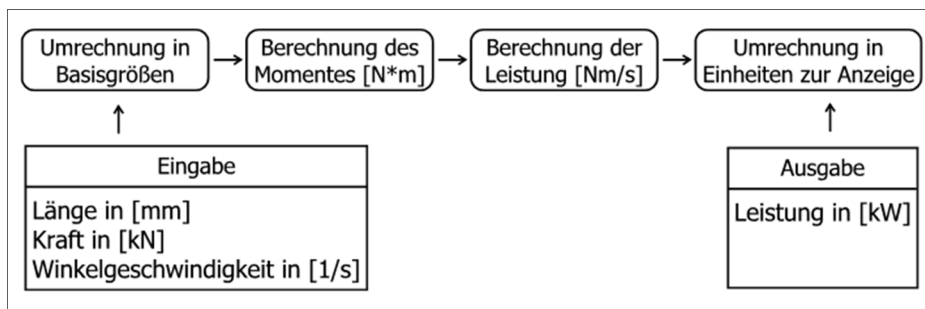


Abbildung 63: Berücksichtigung der Einheiten beim Berechnen

Wie in der Abbildung zu sehen ist, läuft die Berechnung immer mit den Größen (Parametern) in Basiseinheiten. Zur Anzeige (Eingabe und Ausgabe) können aber andere abgeleitete oder mit einem Multiplikator versehene Einheiten genutzt werden. Aufgrund des Ausbaus der länderübergreifenden Zusammenarbeit berücksichtigt die in Tabelle 8 dargestellte Umrechnungstabelle auch das US-amerikanische Einheitensystem.

ID	Einheit	Basiseinheit	Umrechnungsfaktor	System
1	-	-	1	1#2
2	%	%	1	1#2
3	°	°	1	1#2
4	rad	°	57,29578	1#2
5	°min	°	0,016667	1#2
6	°sek	°	0,000278	1#2
7	quad	°	90	1#2
8	°/s	°/s	1	1#2
9	rad/s	°/s	57,29578	1#2
...	...	...	...	...
14	‰	‰	1	1#2
15	1/mm	1/mm	1	1
16	1/yd	1/mm	0,001094	2
...	...	...	...	...

Tabelle 8: Umrechnungstabelle der Einheiten (System 1 – metrisch, System 2 – US-amerikanisch)

### 3.2 Ermittlung von Merkmalswerten

Die Bestimmung der Merkmalswerte kann über zwei Wege erfolgen. Sie können entweder in den Anforderungen als Zielgrößen festgelegt oder als Ergebnisse verschiedener Berechnungen bestimmt werden. Unter einer elementaren Berechnung versteht sich eine Funktion (Umwandlung) der Eingangsparameter in Ausgangsparameter mit Hilfe einer bestimmten Regel (siehe Abbildung 64).

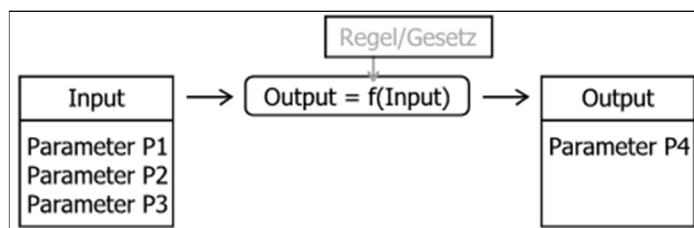


Abbildung 64: Elementare Berechnung



Regeln bzw. Gesetze können dabei verschiedener Natur sein, zum Beispiel geometrisch, physikalisch usw. Die einzelnen Funktionen für die Ermittlung der Merkmalswerte sind dabei sehr stark miteinander vernetzt, so dass die Ausgabeparameter einer Funktion oft Eingabeparameter einer anderen Funktion sind (siehe Abbildung 65).

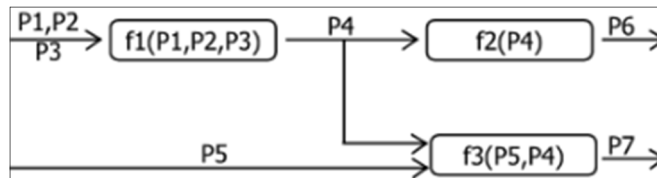


Abbildung 65: Graphische Darstellung der Vernetzung von Berechnungsfunktionen

Eine Baugruppenberechnung beinhaltet oft sehr viele einzelne Teilberechnungen, welche stark miteinander vernetzt sind. Für viele dieser Teilaufgaben existieren in der Regel bewährte Lösungen, welche in Form von Normen, Richtlinien oder betriebsinternen Vorschriften, oft noch in Papierform, vorliegen. Ein einfaches Beispiel dafür, ist die Berechnung des Achsabstandes einer Stirnradstufe in einem Getriebe. Diese Berechnung ist in DIN 3960:1987-03 folgendermaßen definiert:

$$a = a_d \cdot \frac{\cos(\alpha_t)}{\cos(\alpha_{wt})} = \frac{m_n \cdot (z_1 + z_2)}{2 \cdot \cos(\beta)} \cdot \frac{\cos(\alpha_t)}{\cos(\alpha_{wt})} \quad (3-1)$$

mit

$$a_d = \frac{d_1 + d_2}{2} = m_t \cdot \frac{z_1 + z_2}{2} = \frac{m_n \cdot (z_1 + z_2)}{2 \cdot \cos(\beta)} \quad (3-2)$$

Hierbei bedeuten  $a$  – Achsabstand in mm,  $a_d$  – Null-Achsabstand in mm,  $m_n$  – Normalmodul in mm,  $m_t$  – Stirnmodul in mm,  $z_{1,2}$  – Anzahl der Zähne vom Ritzel und Rad,  $d_{1,2}$  – Teilkreisdurchmesser vom Ritzel und Rad in mm,  $\beta$  – Schrägungswinkel in °,  $\alpha_t$  – Stirneingriffswinkel in °,  $\alpha_{wt}$  – Betriebseingriffswinkel in °.

Unter Berücksichtigung des Aufbaus einer Funktion nach dem Grundprinzip der Datenverarbeitung (siehe Abbildung 66) ist es möglich, zu mehreren Funktionen mit unterschiedlichen Eingangs- und Ausgangsgrößen zu kommen.

Beschreibung der Funktion		
Eingabe (Parameter)	Transformation	Ausgabe (Parameter)

Abbildung 66: Funktionsaufbau nach dem EDV-Prinzip

Abhängig von den Anfangsbedingungen (bekannten Größen) kann die Ermittlung des Achsabstandes eines Zahnradpaares durch unterschiedliche Kombinationen der in Tabelle 9 und Tabelle 10 aufgeführten Funktionen (Bausteine) erfolgen.

Bezeichnung	Eingangsgrößen	Ausgangsgrößen	Transformation
F1	$d_1, d_2$	$a_d$	$(d_1 + d_2)/2$
F2	$m_t, z_1, d, z_2$	$a_d$	$m_t * (z_1 + z_2)/2$
F3	$m_n, \beta, z_1, z_2$	$a_d$	$m_n * (z_1 + z_2)/(2 * \cos(\beta))$

Tabelle 9: Funktionen für die Ermittlung des Wertes von „ $a_d$ “

Bezeichnung	Eingangsgrößen	Ausgangsgrößen	Transformation
F11	$a_d, \alpha_t, \alpha_{wt}$	$a$	$a_d * \cos(\alpha_t) / \cos(\alpha_{wt})$
F12	$m_n, z_1, z_2, \alpha_t, \alpha_{wt}$	$a$	$m_n * (z_1 + z_2) / (2 * \cos(\beta)) * \cos(\alpha_t) / \cos(\alpha_{wt})$

Tabelle 10: Funktionen für die Ermittlung des Wertes von „ $a$ “

### 3.2.1 Funktionsbaustein als elementar funktionales Prozesselement

In der vorliegenden Arbeit wird zur Präzision der Sprachregelungen anstatt des relativ abstrakten Begriffes „Funktion“ der Begriff „Funktionsbaustein“ eingeführt.

Der Begriff Funktion beschreibt in der Konstruktion (Pahl und Beitz, 1997) den gewollten Zusammenhang zwischen Eingangs- und Ausgangsgrößen eines Systems mit dem Ziel, eine bestimmte Aufgabe zu erfüllen.

Funktionsbaustein – Dieser Begriff wurde analog zum „Prozessbaustein“ (VDI 2206:2004) eingeführt:

„Ein Prozessbaustein ist eine abgeschlossene Einheit von Tätigkeiten, die dazu dienen, ein bestimmtes Zwischenziel zu erreichen. Ein Funktionsbaustein enthält darüber hinaus eine Beschreibung der Eingangs- und Ausgangsinformationen, Klassifizierungskriterien und Zusatzinformationen (wie unterstützende Methoden oder erforderliche Kompetenzen), die das Arbeiten mit dem Prozessbaustein unterstützen“ (VDI 2206:2004). Er beschreibt auch wieder auftretende vordefinierte Teilschritte bei der Auslegung oder Nachrechnung von Maschinenelementen sowie deren Dokumentation. Im Gegensatz zu den sich auf der Makroebene befindenden Prozessbausteinen beschreibt ein Funktionsbaustein eine elementare und damit auf der Mikroebene agierende vordefinierte Funktion wie zum Beispiel die Berechnungsfunktion für die Ermittlung des Null-Achsabstandes einer Zahnradstufe nach DIN 3960:1987-03 (siehe Tabelle 9). Somit repräsentieren die Funktionsbausteine elementare Berechnungen zur Ermittlung von geometrischen oder physikalischen Eigenschaften eines Objektes bzw. einer Baugruppe von Maschinenelementen. Diese Eigenschaften sind auch gleichzeitig die Koppelstellen zwischen den Funktionsbausteinen. Würde der komplette Inhalt der DIN 3960:1987-03 bzw. der neuen DIN ISO 21771:2014-08 modularisiert werden, könnte daraus eine Bibliothek der aufeinander abgestimmten Funktionsbausteine für die Berechnung der Verzahnungsgeometrie entstehen. Mit einer analogen Umsetzung der anderen Normen, wie zum Beispiel der DIN 3990 und

der ISO 6336, ergäbe sich eine umfangreiche Wissens-Bibliothek als Grundlage für die Verzahnungsberechnung.

### 3.2.2 Datenaustausch zwischen den Funktionsbausteinen

Die Berechnung einzelner Maschinenelemente besteht aus elementaren Berechnungsroutinen. Diese formt sich durch eine gerichtete Abfolge von Schritten, welche bestimmten logischen, mathematischen, geometrischen oder physikalischen Zusammenhängen unterliegen. Diese Abfolge kann mit Hilfe von Aktivitätsdiagrammen dargestellt werden. In der Abbildung 67 ist das Beispiel für die Berechnung des Achsabstandes eines Stirnradpaares dargestellt.

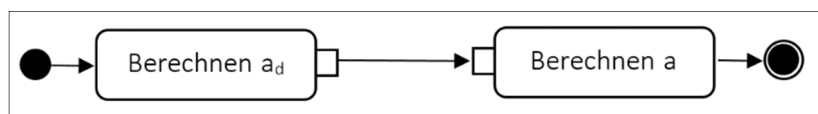


Abbildung 67: Einfaches SysML-Aktivitätsdiagramm mit zwei Aktionen

Dabei wird nur der Wert des Null-Achsabstandes  $a_d$  nach der Berechnung als Eingabe für die Funktion der Berechnung des Achsabstandes bereitgestellt. Diese Bereitstellung kann mit Hilfe eines Pins vom Typ Achsabstand erfolgen. Pins einer Aktion in SysML stellen Verbindungspunkte zu dieser Funktion für andere Funktionen dar. In Bezug auf die Berechnungsfunktion repräsentieren Pins die Parameter einer Funktion sowie deren Ergebnis. Zwei Pins vom selben Typ können mit Hilfe einer gerichteten Kante verbunden werden, wenn ein Pin ein Eingabe-Pin und der andere ein Ausgabe-Pin sind. Dabei beschreibt die Kante die Richtung des Datenflusses. Diese Art des Datenaustauschs wird direkter Datenaustausch in einer datenflussorientierten Abfolge von Funktionen genannt. Da die Datenflüsse in einem SysML-Aktivitätsdiagramm explizit modelliert werden müssen, ist die Abbildung von einfacheren sequentiellen Berechnungen sehr intuitiv. Es müssen lediglich die entsprechenden Ausgabeparameter eines Funktionsbausteines mit den Eingabeparametern des anderen Funktionsbausteines verbunden werden.

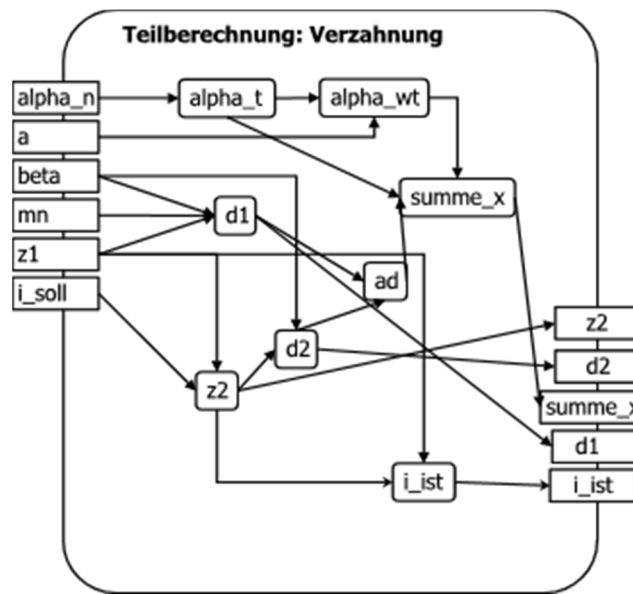


Abbildung 68: Beispiel der Verzahnungsberechnung mit Aktivitätsdiagramm

Die Anwendung dieses Prinzips auf die Modellierung eines umfangreich verzweigten Berechnungsprozesses zeigt deutlich, dass die datenflussorientierte mehrfache Kopplung eines Funktionsbausteins mit den anderen Modellelementen schnell zu unübersichtlichen Darstellungen des gesamten Prozesses, wie zum Beispiel auf der Abbildung 68 zu sehen ist, führen kann. Abhilfe kann eine kontrollflussorientierte Modellierung der Prozesse schaffen, bei der nicht mehr der Datenfluss modelliert wird, sondern die Richtung der Abarbeitung der Prozessbausteine. Der Datenaustausch zwischen den prozessbeteiligten Elementen soll dabei aber automatisch ablaufen.

#### Kooperation vs. Kommunikation zwischen den Prozessbeteiligten

Zu den üblichen Ansätzen für ein automatisiertes Datenmanagement gehören zum Beispiel das Dokumentenmigrationsprinzip oder das Prinzip des gemeinsamen Datenspeichers. Mit deren Hilfe kann die Transparenz der Abläufe stark gesteigert werden. Bei dem Dokumentenmigrationsprinzip wird der Datenfluss mit dem Sequenzfluss zusammengelegt, indem das Dokument entsprechend dem Kontrollfluss von einer Aktivität zur anderen weitergeleitet wird. Am besten vorstellen kann man sich dieses Prinzip als eine Dokumentenmappe, die von einem Bearbeiter zum nächsten weitergereicht und mit entsprechenden Informationen ergänzt wird (siehe Abbildung 69) (Lütke-meier und Thöne, 2001)

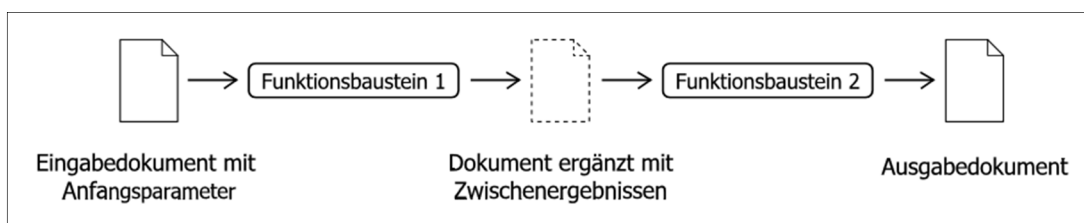


Abbildung 69: Dokumentenmigrationsprinzip

Umgesetzt auf die Berechnungsbausteine bedeutet dies, dass jeder Bearbeiter für die eigene Ausführung alle zu diesem Zeitpunkt bekannten Informationen (Parameter) erhält bzw. zur Verfügung hat und mit den eigenen Aktivitätsergebnissen ergänzt. Somit beinhaltet das Dokument am Ende der Abarbeitung des Prozesses alle Eingangsdaten sowie alle berechneten Werte.

Bei der Verwendung eines zentralen Speichers wird von einem indirekten Datenaustausch gesprochen. Die Funktionsbausteine erhalten nur einen Verweis auf die notwendigen Parameter, welche in dem gemeinsamen Speicher zu finden sind (siehe Abbildung 70). Der zentrale Speicher wird in den meisten Fällen als Datenbank realisiert, kann aber genauso ein Dokument sein.

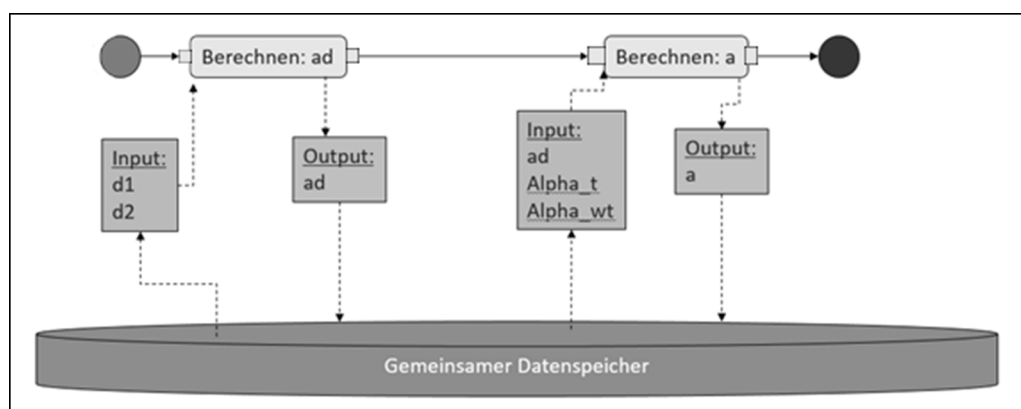


Abbildung 70: Prinzip des gemeinsamen (zentralen) Speichers

Die Datenflüsse werden dabei nicht explizit modelliert, sondern ergeben sich automatisch durch Zugriffe der Funktionsbausteine auf die Merkmale (Prozessvariablen) (siehe Abbildung 71). Ein lesender bzw. schreibender Zugriff auf ein Merkmal M durch einen Funktionsbaustein FB1 ist dann gegeben, wenn M als aktueller Eingabe- bzw. Ausgabeparameter in der durch FB1 realisierten Schnittstelle vorkommt. Wird M als aktueller Ausgabeparameter von Funktionsbaustein FB1 und als aktueller Eingabeparameter von Funktionsbaustein FB2 verwendet, so existiert ein Datenfluss von FB1 nach FB2 (Wörzberger, 2010)

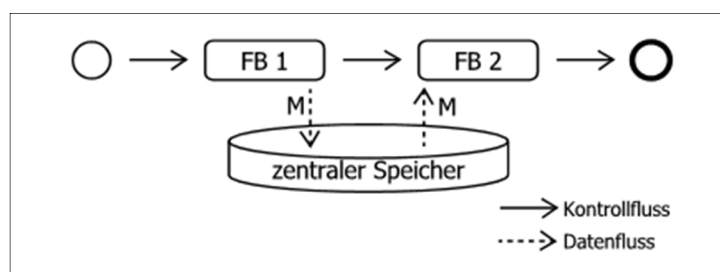


Abbildung 71: Datenfluss zwischen den Funktionsbausteinen

Die beiden Ansätze „Dokumentmigrationsprinzip“ und „Verwendung eines zentralen (gemeinsamen) Speichers“ entsprechen den Konzepten Kommunikation (direkter Datenaustausch zwischen den Komponenten) und Kooperation (indirekter Datenaustausch zwischen den Komponenten) zum

Datenaustausch im Bereich der verteilten Systeme und sind damit als gleichmächtig zu bezeichnen (Lütke-meier und Thöne, 2001). Im Rahmen der vorliegenden Arbeit wurde die Kooperation zwischen den Funktionsbausteinen und damit der indirekte Datenaustausch nach dem Prinzip des gemeinsamen (zentralen) Datenspeichers implementiert.

### 3.2.3 Modellierung von Berechnungsprozessen

Im Vergleich zum direkten Datenaustausch (Kommunikation) zwischen den Funktionsbausteinen im Beispiel der Verzahnungsberechnung bringt die Anwendung des Prinzips des gemeinsamen Speichers eine deutliche Verbesserung der Lesbarkeit des Ablaufes mit sich (siehe Abbildung 72)

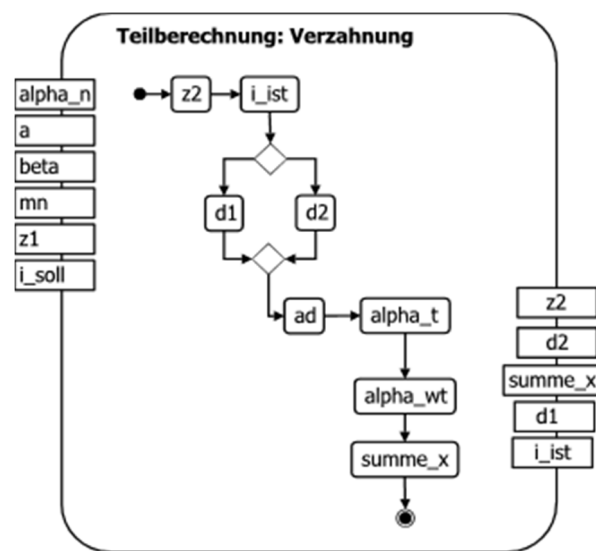


Abbildung 72: Beispielprozess

Damit die Abarbeitung der Prozesskette reibungslos abläuft und das Modell nicht von den Fach- und Sprachkenntnissen des Bearbeiters beeinflusst wird, müssen die Schnittstellen zwischen den Funktionsbausteinen eindeutig definiert sein. Ein ungehinderter und fehlerfreier Datenfluss zwischen den Funktionsbausteinen ist nur dann möglich, wenn die Typen der zu übertragenden Informationselemente (Pins) am Ausgang des sendenden und am Eingang des empfangenden Funktionsbausteins übereinstimmen. In SysML können für die Pins alle denkbaren Datentypen definiert werden, angefangen von den elementaren Zeichenketten (String) und den ganzen Zahlen (Integer) bis hin zu den Klassen und anderen komplexen Datenstrukturen sowie realen Objekten (Alt, 2012).

Die Anwendung der Klassifizierung auf Produkteigenschaften und damit auf die Schnittstellen (Pins) der Funktionsbausteine ist in der Abbildung 73 veranschaulicht.

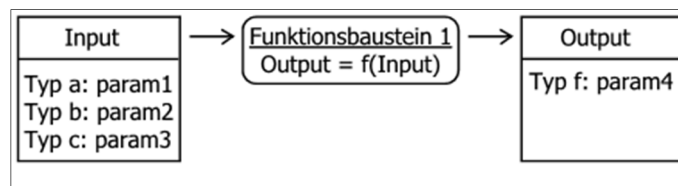


Abbildung 73: Funktionsbaustein

Hierbei entsprechen die Typen der Eingabe- und Ausgabeparameter den Merkmalen der Produktklassifizierung nach dem eCl@ss-Standard.

Bei den Funktionsbausteinen selbst bedarf es keiner zusätzlichen Klassifizierung, denn sie werden der Untergruppe (Klasse) des Ergebnis-Merkmals zugeordnet (siehe z. B. Tabelle 9 und Tabelle 10). Zur einheitlichen Sprachregelung wird weiterhin anstatt der allgemeingültigen Bezeichnung „Untergruppe“, der übliche Begriff „Klasse“ verwendet. In der Abbildung 74 ist der hierarchische Aufbau der Datenstruktur zusammengefasst.

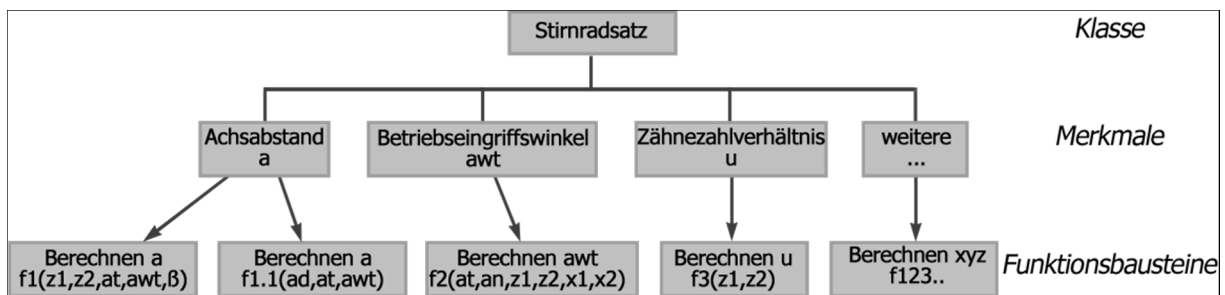


Abbildung 74: Hierarchische Darstellung der Aufbaustruktur

### 3.2.4 Formalisierung der Funktionsbausteine

Damit ein Funktionsbaustein eindeutig interpretiert werden kann, muss er eine eindeutige Semantik und Syntax aufweisen. Die graphische Darstellung eines Funktionsbausteins resultiert aus der BPM Notation, in der die Art der Aktivitäten durch ein zusätzliches Symbol gekennzeichnet wird. Dies kann beispielsweise eine manuelle Aufgabe, ein auszuführendes Skript oder ein Teilprozess sein (siehe Abbildung 75).



Abbildung 75: Graphische Darstellung einiger Modellierungselemente in BPMN

Da die meisten Funktionsbausteine automatisiert abgearbeitet werden sollen, werden diese wie eine Skript-Aktivität dargestellt. Die manuellen Eingriffe innerhalb des Prozesses, wie zum Beispiel die Auswahl eines Wertesatzes aus einer Datenbank, werden als eine manuelle Aufgabe mit nach rechts zeigender Handfläche gekennzeichnet. Ein Teilprozess kennzeichnet eine zusammenhängende Abfolge von Aktivitäten und wird näher in Kapitel 3.3.2 betrachtet. Die minimal notwendigen Eigenschaften

eines Funktionsbausteins resultieren aus der Definition einer Aktivität in (OMG, 2011, OMG, 2015a, OMG, 2015b):

- ParentAttribute: ID des Ergebnismerkmals
- Name: Kurzbezeichnung als Text
- InputSet: Menge der Eingabeparameter als Liste von Merkmalen
- Body: Programmcode / Skript zur Ausführung der Berechnungsfunktion
- Quelle: Bezeichnung der Quelle für die Berechnungsfunktion als Text
- Formel: mathematische Beschreibung der Funktion als Bild oder Text

Definition der auszuführenden Funktion in einem Funktionsbaustein

Im Rahmen der vorliegenden Arbeit wurde für das Management der Berechnungsprozesse und deren Ausführung der Prototyp eines Autorensystems entwickelt. Das Ziel dieses Autorensystems ist die Bereitstellung der modular aufgebauten Wissens Elemente und deren intuitive Verknüpfung zu einem Prozess.

In den letzten Jahren wurde das Thema „Zero-Coding“ besonders im Bereich der Geschäftsprozessmodellierung sehr stark vorangetrieben. Dabei wird angestrebt, die Entwicklungs-umgebung mit Konfigurationsdialogen so zu unterstützen, dass ein Prozessmodell entsteht, welches ohne Einsatz der klassischen Programmierung in ein ausführbares technisches Modell überführt werden kann. Die dabei zur Verfügung gestellten Bausteine (Services, Adapter, etc.) werden so ohne weiteren Aufwand miteinander verbunden. Zusätzliche Arbeit muss dann geleistet werden, wenn die gewünschte Funktionalität im Programm nicht als Baustein zur Verfügung steht. Außerdem kann die Fülle der möglichen Szenarien so groß sein, dass die Entwicklung eines Systems nach dem Zero-Code Ansatz deutlich komplexer und kostspieliger wird, als die Vorgehensweise bei der klassischen Programmierung. Dies führte dazu, dass der „Less-Code“ Ansatz große Verbreitung fand. Dabei werden möglichst viele Programmierschritte über die Konfigurationsdialoge abgewickelt. Die Definitionen eines neuen Bausteins oder einer Datenschnittstelle erfolgten jedoch unter Verwendung klassischer Programmiersprachen (Signavio, 2012).

Im Rahmen der hier vorliegenden Arbeit wird bei der Realisierung des Autorensystems der „Less-Code“ Ansatz verfolgt. Wobei nur die Definition der Übertragungsfunktion eines Funktionsbausteins bzw. eines Kontrollknotens zwischen den Input- und Outputparametern mit einer Programmiersprache erfolgt, da die Ausführung der mathematischen Funktionen bzw. der Algorithmen in den Funktionsbausteinen bereits vorab verankert wurden.

Die Wahl der Programmiersprache ist in erster Linie von der Handhabung und Erlernbarkeit der Sprache abhängig. Dabei soll berücksichtigt werden, dass das zu entwickelnde System zur Berechnung



von Maschinenelementen und deren Baugruppen auf Ingenieure ausgerichtet ist und nicht auf Programmierer. Die Funktionen in den Funktionsbausteinen sollen einerseits durch die vorgenommene Modularisierung sehr einfach gehalten werden, andererseits erfordert das Autorensystem im Hinblick auf den Umfang der Skripte die Möglichkeit, auch komplexere Algorithmen und Schnittstellen als Funktionsbausteine implementieren zu können.

Für die Funktionsbausteine wurde im Rahmen der Arbeit die Programmiersprache C#( C Sharp) ausgewählt. Dies ist eine moderne und sehr verbreitete objektorientierte Sprache, die alle gestellten Voraussetzungen erfüllt. Sie ist leicht zu erlernen, zukunftsicher und setzt nahezu keine Grenzen für die Kreativität bei der Erstellung von Funktionsbausteinen. Zur genaueren Beurteilung der ausgewählten Programmiersprache soll an dieser Stelle auf die Literatur: (Henning und Vogelsang, 2006, Büchner, 2007) verwiesen werden.

#### Infosystem eines Funktionsbausteines

Zur vollständigen Kennzeichnung eines Funktionsbausteins gehören neben der eindeutigen Identifikation auch die informativen Angaben, welche dabei helfen, den Baustein nicht nur schnell wiederzufinden, sondern auch anwenden zu können. Es werden deshalb, analog zu den Sachmerkmalen im eCl@ss-System, zu jedem Funktionsbaustein die Beschreibungen und Angaben zum fachlichen Zusammenhang abgelegt. Hierfür wurde ein Informationssystem der Funktionsbausteine entwickelt, das insbesondere Aspekte wie Einheitlichkeit, Internationalität sowie Pflege und Weiterentwicklung berücksichtigt.

Als computerbasierte Sprache für die Informationsaufbereitung wird zunächst die Hypertext Markup Language (HTML) gewählt, um die Wissensmodule möglichst hardwareunabhängig gestalten zu können. Dies hat den Vorteil, dass die Informationsobjekte auf unterschiedlichen Endgeräten und in unterschiedlichen Systemumgebungen genutzt werden können. Die beschreibenden Eigenschaften der Wissensmodule eines Funktionsbausteines wurden bereits aufgelistet. Davon werden für das Hilfesystem folgende Eigenschaften benötigt:

- Kurzbezeichnung,
- Symbol,
- Beschreibung,
- Quelle und
- Angaben zu dem mathematischen Ausdruck bzw. dem Algorithmus im Ausführungsskript.

Diese letzte Angabe kann entweder textuell oder mit Hilfe einer Graphik (z. B. als Screenshot im PNG oder JPEG Format) erfasst werden. Eine Besonderheit bei der textuellen Definition besteht darin, dass diese auch in der Mathematical Markup Language (MathML) (W3C, 2014b) Sprache erfolgen kann.

Dieses Dokumentenformat erlaubt die Beschreibung und anschließende Anzeige von mathematischen Formeln und komplexen Ausdrücken im Internet. Das Format ist mit ISO/IEC 40314:2016-03 standardisiert und ein Bestandteil von HTML5 (W3C, 2014a).

### 3.2.5 Strukturierung der Merkmale und Funktionsbausteine

Die Zuordnung der Funktionsbausteine zu den Merkmalen erfolgt nach ihrem Ergebnismerkmal. Es können abhängig von der Quelle und den Beziehungen zu den anderen Produkteigenschaften mehrere Funktionsbausteine einem Merkmal zugeordnet sein. Anders ausgedrückt, der Wert eines Merkmals kann häufig durch mehrere unterschiedliche Gleichungen bestimmt werden. Bei den Klassen, welche komplexe Bauteile oder Baugruppen repräsentieren, wie zum Beispiel Wellen, Zahnräder oder Zahnradpaare kann die Merkmalsleiste sehr umfangreich sein. Die Anzahl der Funktionsbausteine wird dementsprechend groß. Die Klasse „Zahnrad“ hat beispielsweise Dutzende von verschiedenen Funktionsbausteinen. Vor allem, um die Suche nach dem passenden Funktionsbaustein bei der Modellierung eines Berechnungsprozesses zu erleichtern, wäre es sinnvoll, diese aufgabengerecht zu strukturieren. Besonders wichtig ist die Klassenzuordnung der fachspezifischen Merkmale, da bei interdisziplinären Aufgabenstellungen die Bezeichnungen der bereits vorhandenen Merkmale auf den anderen Informationsfeldern nicht immer im Voraus bekannt sind. Dabei soll möglichst keine weitere Ebene im Klassifikationssystem eingeführt werden, um einen direkten Umgang mit den Funktionsbausteinen zu ermöglichen.

Eine allgemein gehaltene Strukturierung könnte die Merkmale beispielsweise nach der Zugehörigkeit zu den Parametergruppen „Geometrie“, „Belastung“, „Werkstoffe“ oder „Umgebung“ ordnen. Die unter diese Begriffe fallenden Eigenschaften und Randbedingungen bilden bei Maschinenelemente-Berechnungen wichtige Hauptgruppen. Werden zum Beispiel weitere Unterteilungen vorgenommen, zum Beispiel nach Stirn- und Breitenlastfaktoren bei einem Zahnradpaar, sind diese nicht mehr allgemeingültig und beziehen sich nur noch auf die entsprechende Klasse. Ein weiteres Beispiel hierfür ist die Klasse „Zugfeder“, in der ebenfalls spezielle Merkmalsgruppen vorkommen können, wie zum Beispiel die „Konstruktive Gestaltung“, in der die Lagerung der Federenden und andere Merkmale zur Beschreibung der Ösen und Federwindungen zu finden sind.

Für die Definition solcher zusammenhängender Parametergruppen existieren in eCl@ss Advanced Strukturelemente wie Aspekte und Merkmalsblöcke. „Als Block bezeichnet man die Sammlung verschiedener klassenbezogener, zusammenhängender Merkmale unter einem Namen. Bei umfangreichen Beschreibungen von Geräten mit Klassen ist eine solche Strukturierung äußerst hilfreich. Zur Erstellung eines Blocks muss in der Advanced-Repräsentation ein Referenzmerkmal gebildet werden“ (eCl@ss e.V., o. D.). Als Beispiel wird folgende Struktur angeboten:

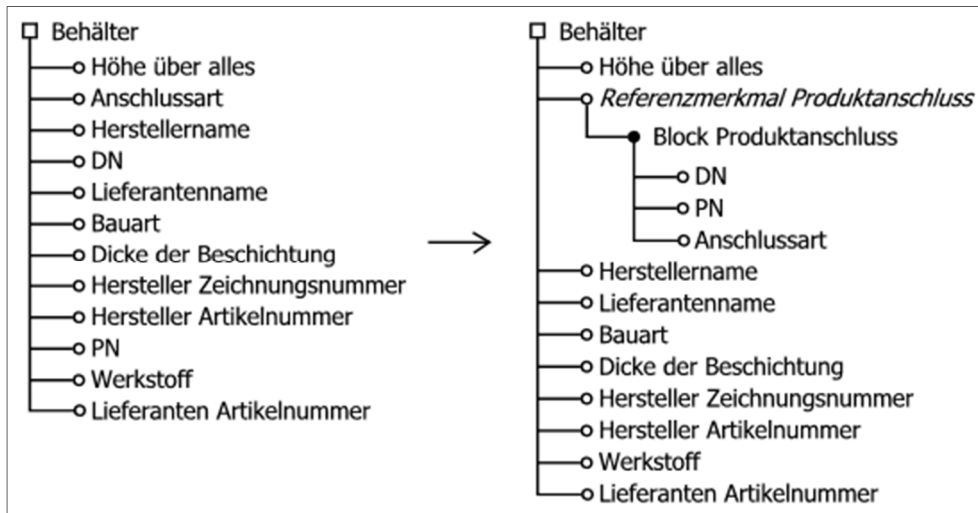


Abbildung 76: Verwendung von Merkmalsblöcken in eCl@ss (Uiterwyk, 2012)

Die Block-Definition unterscheidet sich jedoch von der in der DIN 4002 angegebenen Form. Die Norm DIN 4002, welche die eigentliche Grundlage der Produktklassifizierung in eCl@ss darstellt (Institut der deutschen Wirtschaft Köln, 2010), liefert eine etwas allgemeinere Definition des Merkmalblocks:

„Zusammenfassung von Merkmalen eines Teiles eines Objektes (z. B. Logistik-, Geometrie-, Funktions- oder Formaspekte; Werkstoffangaben, -anforderungen oder -eigenschaften), die nicht unabhängig vom eigentlichen Objekt existieren können, sich aber jeweils strukturiert durch ein Konzept und zugeordnete Merkmale beschreiben lassen“ (DIN 4002-2:2007-07). Als beispielhafte Darstellung der Strukturierung von Merkmalsleisten mit Hilfe von Merkmalsblöcken ist in DIN 4002-2 folgendes Konstrukt aufgeführt:

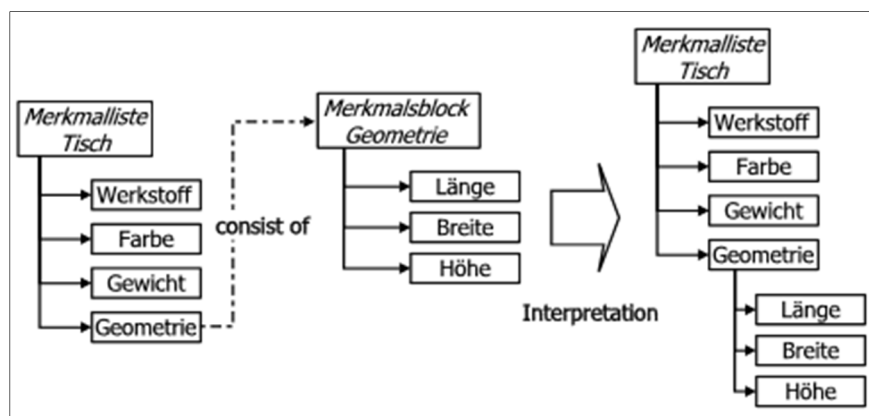


Abbildung 77: Beispiel der Anwendung von Merkmalsblöcken (vgl. ebd.)

Für die Zwecke der Bündelung beziehungsweise Strukturierung der Merkmale nach „Geometrie“, „Randbedingungen“ oder „Umgebung“ zur Erhöhung der Übersichtlichkeit der Merkmalsleisten ist der allgemeinere Ansatz deutlich vorteilhafter, da keine weiteren Klassifikationselemente und Referenzmerkmale gebildet werden müssen. Bei Bedarf ist der spätere Umstieg auf die eCl@ss

Definition problemlos möglich. Die Anwendung dieses Ansatzes auf die Klasse Stirnrad liefert folgende Strukturierung der Merkmale und der Funktionsbausteine (siehe Abbildung 78):

Merkmalsleiste der Klasse: Stirnrad (eCl@ss: 23 - 17 - 01 - 01)		Funktionsbausteine der Klasse: Stirnrad (eCl@ss: 23 - 17 - 01 - 01)	
<u>Geometrie</u>		<u>Geometrie</u>	
Modul	$m_n$	$d = f(z, m_n, \beta)$	
Anzahl der Zähne	$z$	$d_a = f(d, z, h_a)$	
Teilkreisdurchmesser	$d$	$d_b = f(z, m_n, \alpha_n, \beta)$	
Kopfkreisdurchmesser	$d_a$	$h_a = f(m_n, h_{aP^*}, x, k)$	
Zahnbreite	$b$	$s_n = f(m_n, x, \alpha_n)$	
...		$d_w = f(d_b, \alpha_{wt})$	
<u>Werkstoff</u>		$h_f = f(h_{fP}, x, m_n)$	
Zahnfußfestigkeit	$\sigma_F$	...	
Zahnflankenfestigkeit	$\sigma_H$	<u>Tragfähigkeit</u>	
Oberflächenhärte	HB / HV	$\sigma_{F0} = f(F_t, b, m_n, Y_F, Y_S, Y_{\alpha}, Y_B, Y_{DT})$	
Elastizitätsmodul	E	$\sigma_{Fa} = f(\sigma_{F0}, K_A, K_V, K_{Fa}, K_{Fa})$	
...		$S_F = f(\sigma_{FG}, \sigma_{Fa})$	
<u>Allgemein</u>		...	
Verzahnungsqualität			
...			

Abbildung 78: Auszug aus den Merkmalsleisten mit Merkmalsgruppen

Die Eingabe der Daten (Merkmale und entsprechende Funktionsbausteine) in die Wissensdatenbank erfolgt bis jetzt manuell über die entsprechenden GUI-Interfaces. Eine digitale Bereitstellung der genormten Merkmale wurde schon seitens DIN unternommen, indem 2003 ein DIN-Merkmallexikon bzw. ein DIN-Merkmalserver als Pilotprojekte entstanden. Der DIN-Merkmalserver sollte die Inhalte der Normreihen DIN 4000 einfacher zugänglich machen und gleichzeitig die Basis für andere Merkmalssysteme und Klassifikationen bilden (Institut der deutschen Wirtschaft Köln, 2010). Der Fortschritt der Freigabe von Merkmalen lief über Jahre allerdings nur zögerlich voran. Ende 2008 waren es lediglich knapp über 400 Merkmale, die offiziell zur Nutzung freigegeben wurden. Zu diesem Zeitpunkt hatte aber die Wirtschaft schon erhebliche Investitionen im Bereich der Klassifizierung von Produkten, Stammdatenpflege und elektronische Marktplätze unternommen (Miehe, 2008) und entwickelte auf der Basis der DIN 4000 das erweiterte Klassifikationssystem eCl@ss.

Ein weiterer Fortschritt in Richtung der Digitalisierung der Produktentwicklung ist die Bereitstellung von Normdokumenten im XML (eXtensible Markup Language, ein Standard des World Wide Web Consortiums - W3C) (W3C, 2006) Format. Denn im Jahr 2015 hat die DIN-Gruppe begonnen die Normdokumente in das neue Format umzuwandeln. Das Ziel dieses Projektes ist es eine zentrale Datenbank mit dem im XML-Format abgelegten Content der Normen aufzubauen, um diese Dokumente dann in verschiedenen Verwertungs- und Produktformen weiter zu verwenden (DIN, 2016). Wie weit die Klassifizierung und Strukturierung des Contents der Normen in das neue Format

einfließen wird, ist noch unbekannt. Aber allein die Tatsache, dass es digital verfügbar sein wird, eröffnet die Möglichkeiten einer zumindest teilautomatisierten Wissensverarbeitung und Nutzung.

### 3.3 Wiederverwendung von Konzepten

Die Wiederverwendung von Konzeptlösungen im Entwicklungsprozess steigert die Qualität und senkt die Prozesskosten. Dies kann vor allem durch eine hohe Modularisierung des Produktes und deren Baugruppen sowie durch die Steigerung der Transparenz der angewendeten Konzeptlösungen erreicht werden. Die Ansätze zum nachhaltigen Umgang mit konstruktiven Lösungen können in Form von Konstruktionskatalogen (Roth, 1994, Koller, 1994, Leemhuis, 2005) oder Bauteilbibliotheken, die sich heute meist auf Dokumentenebene in den PDM/PLM – Systemen befinden, abgelegt werden. Die in Kapitel 2.2 genannten Konstruktionskataloge bieten zwar die Möglichkeit zur Dokumentation und Wiederverwendung von Lösungswissen, enthalten aber nur wenig informationstechnische Unterstützung. Darüber hinaus liegen sie meist nur in Papierform vor. Diese Tatsache erschwert das Wiederauffinden bereits bewährter konstruktiver Lösungen. Im Gegensatz dazu ist das Wiederfinden von CAD-Daten deutlich einfacher, denn dies wird durch Klassifikationssysteme und Schlagwortverzeichnisse sowie durch die Verwendung geometriebasierter Sachmerkmal-Leisten unterstützt (Leemhuis, 2005, Eigner und Stelzer, 2009). Im Rahmen der hier vorliegenden Arbeit werden mehrere Wissensbibliotheken aufgebaut, angefangen von den Merkmalen der Anforderungen bis zu den Teillösungen im Bereich der Maschinenelementeberechnungen.

#### 3.3.1 Aufbau und Nutzung von Bibliotheken

Dank der Nutzung des Klassifikationsstandards eCl@ss ist der hierarchische Aufbau der Gruppen, Klassen und Merkmale als vorgegeben zu betrachten. Eine der Voraussetzungen für die erfolgreiche praktische Nutzung des im Rahmen der Arbeit vorgestellten Ansatzes ist das Vorhandensein einer gut gefüllten Merkmalsdatenbank. Wie schon in Kapitel 3.1.5 angedeutet, beinhaltet der eCl@ss – Standard gegenwärtig nicht den gewünschten Umfang an technischen Merkmalen, um die gestellten Aufgaben der Beschreibung von Berechnungsprozessen von Maschinenelementen zu lösen. Demzufolge wurde als erste Aufgabe die Erweiterung der bestehenden Sachmerkmal-Leisten vorgenommen. Besondere Hilfestellung bietet hier die 4000er Reihe der deutschen Normung (DIN). Im Teil 141 sind zum Beispiel die Merkmale der Wälzfräser für die Evolventenverzahnung aufgeführt. Auf der nachstehenden Abbildung 79 sind das Bezugsprofil des Wälzfräasers und dessen Merkmale dargestellt.

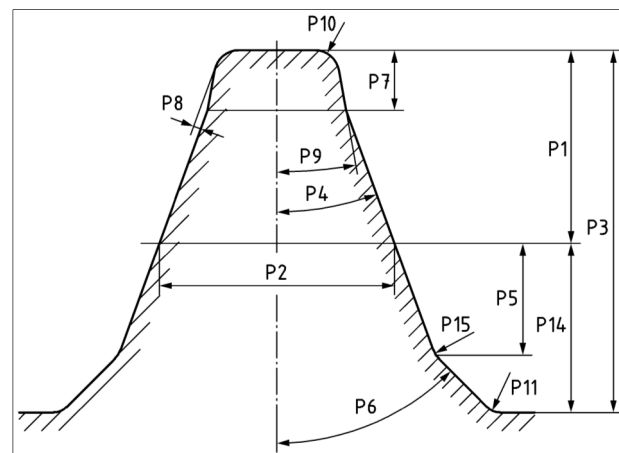


Abbildung 79: Bezugsprofil (Basisprofil) nach DIN 4000-141:2016-10

Die in der Tabelle 11 zusammengefassten die Merkmale beinhalten bereits einige der nötigen Mindestangaben, wie Bezeichnung, Einheit und Format.

Merkmals- Kennung	Merkmalsbenennung	Einheit	Format	Merkmals- Definition
P1	Kopfhöhe	mm	REAL 7.3	—
P2	Zahndicke im Normalschnitt	mm	REAL 7.3	siehe 4.3
P3	Zahnhöhe	mm	REAL 7.3	—
P4	Normaleingriffswinkel	Grad	REAL 10.6	—
P5	Höhe Beginn Kantenbrechflanke	mm	REAL 7.3	—
P6	Kantenbrechflankenwinkel	Grad	REAL 10.6	—
P7	Protuberanzhöhe	mm	REAL 7.3	—
P8	Protuberanzbetrag	mm	REAL 7.3	—
P9	Protuberanzwinkel	Grad	REAL 10.6	—
P10	Kopfrundungsradius	mm	REAL 7.3	—
P11	Fußrundungsradius	mm	REAL 7.3	—
P14	Fußhöhe	mm	REAL 7.3	—
P15	Übergangsradius	mm	REAL 7.3	—
P0	Normalmodul	mm	REAL 8.6	—

Tabelle 11: Auszug aus den Merkmalslisten für Wälzfräser (vgl. ebd.)

Die Kennung der Merkmale ist hier leider dokumentenspezifisch bzw. lokal und kann nicht normenübergreifend genutzt werden. Die Einheit ist als Basiseinheit der im Rahmen dieser Arbeit definierten Merkmalseigenschaften zu verstehen. Die Angabe des Formats wie zum Beispiel „REAL 7.3“ bedeutet eine reelle Zahl mit 7 Vor- und 3 Nachkommastellen (ebd.). Einige Merkmale erhalten im Rahmen der Angabe zur Merkmalsdefinition auch die Informationen zu dem mathematischen

Ausdruck (Formel) zu dessen Wertbestimmung, wie beispielsweise das Merkmal P2, welche die Zahndicke des Rades im Normalschnitt beschreibt:

$$P2 = \frac{P0 \cdot n}{2} \quad (3-3)$$

Dabei ist P0 der Normalmodul. Unter Berücksichtigung der genormten Kurzzeichen ergibt sich aus Gleichung (3-3) folgende Gleichung:

$$s_{P0} = \frac{m_n \cdot n}{2} \quad (3-4)$$

Im Anhang der Norm findet sich die Zuordnung der internen Merkmalskennung zu den Kurzzeichen einer Berechnungsnorm, sofern diese vorhanden sind. In dem vorgestellten Fallbeispiel ist es die Norm (DIN 3960:1987-03) (siehe Tabelle 12 ).

Zuordnung der Merkmalskennungen nach DIN4000-141 zu den Kurzzeichen DIN 3960 Merkmalskennung	Kurzzeichen	Merkmalbenennung
A1	$d_{a0}$	Kopfkreisdurchmesser
P0	$m_n$	Normalmodul
P1	$h_{aP0}$	Kopfhöhe
P2	$s_{P0}$	Zahndicke
P3	$h_{P0}$	Zahnhöhe
P4	$\alpha_{n0}$	Normaleingriffswinkel
P5	$h_{FFP0}$	Höhe, Beginn Kantenbrechflanke
P6	$\alpha_{KP0}$	Kantenbrechflankenwinkel
P7	$h_{prP0}$	Protuberanzhöhe
P8	$pr_{P0}$	Protuberanzbetrag
P9	$\alpha_{prP0}$	Protuberanzflankenwinkel
P10	$\rho_{aP0}$	Kopfrundungsradius
P11	$\rho_{fP0}$	Fußrundungsradius
P12	$p_n$	Normalteilung
P14	$h_{fP0}$	Fußhöhe
R18	$d_f$	Fußkreisdurchmesser

Tabelle 12: Zuordnung der Merkmalskennungen nach DIN 4000-141 zu den Kurzzeichen der DIN 3960:1987-03)

Da diese Daten in die Merkmalsbibliothek manuell übertragen werden müssen, ist diese Tätigkeit mit einem bestimmten Arbeitsaufwand und einer erhöhten Fehleranfälligkeit verbunden. Das gleiche gilt auch für die Funktionsbausteine, welche die elementaren Berechnungselemente repräsentieren und sehr unterschiedlich sind, so dass diese meistens neu angelegt werden müssen. Die Prozesse, besonders innerhalb eines Fachgebietes weisen dagegen oft gemeinsame Teilprozesse auf. Wie solche Prozesse und Teilprozesse organisiert werden, wird im nächsten Kapitel näher erläutert.

### 3.3.2 Sub-Prozesse

Im Produktentstehungsprozess beginnt die Beschreibung eines Berechnungsprozesses nur selten bei den Elementarfunktionen auf der untersten Ebene. Oft sind diese schon vorhanden, haben Standardcharakter und sind zu Sub-Prozessen zusammengefasst. Diese mehrstufigen Beziehungen und Prozesse zur Berechnung von Geometrie, Festigkeit und Lebensdauer bilden in sich abgeschlossene Funktionen und können als solche bestimmten Klassen oder Gruppe von Klassen zugeordnet werden. Ein gutes Beispiel ist die Ermittlung der Flanken- oder Fußtragfähigkeit einer Verzahnung, die der Gruppe „Zahnradpaar“ oder „Verzahnungselement“ zugeordnet werden kann. Im nächsten Kapitel wird auf das Thema Sub-Prozess und dessen Wiederverwendbarkeit näher eingegangen.

Umfangreiche Produktmodelle und damit verbundene Berechnungsprozesse sind oft wenig transparent und kaum nachvollziehbar. Sollte sich der dort abgebildete Sachverhalt verändern oder müssen Anpassungen an Infrastrukturen vorgenommen werden, erfordert dies einen hohen Aufwand. Es ist also ratsam, fachlich zusammenhängende und in sich abgeschlossene Teilprozesse (Sub-Prozesse) zusammenzufassen und auszulagern. Der Umfang solcher Sub-Prozesse richtet sich nach der Komplexität sowie nach der Häufigkeit der Wiederverwendung. In der Abbildung 80 sind zum Beispiel die Lagerberechnung nach DIN 281:2010-10 und die Wellenberechnung nach DIN 743-1:2012-12 so umfangreich, dass diese als Sub-Prozesse dargestellt sind, um den eigentlichen Prozess der Getriebeberechnung hervorzuheben.



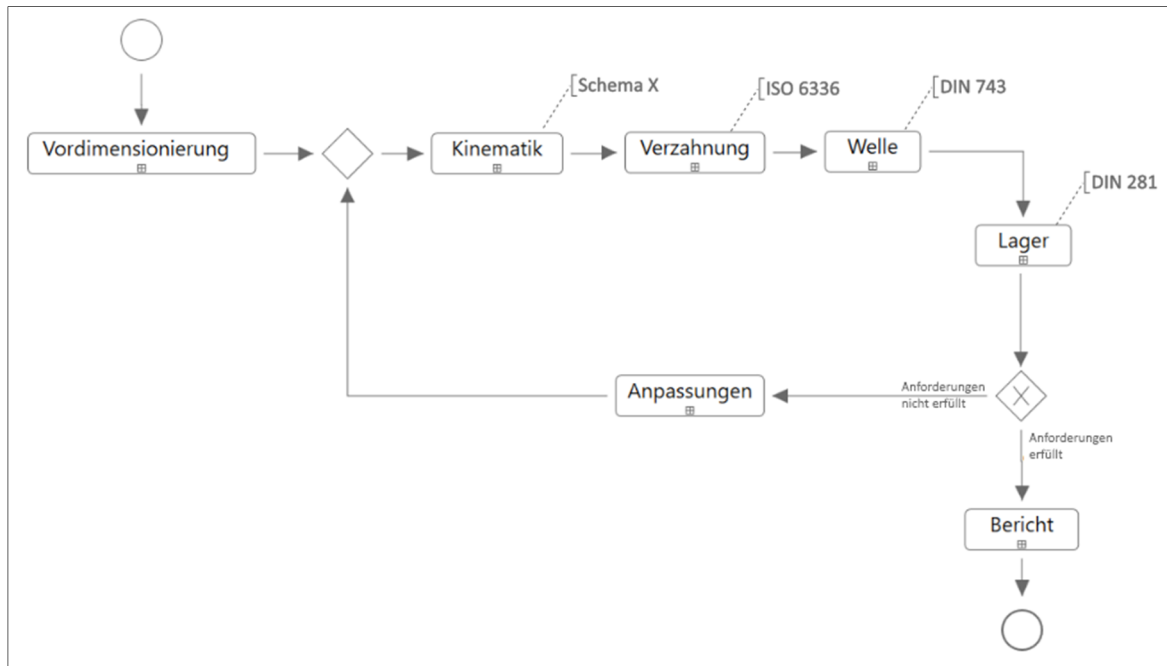


Abbildung 80: Schematischer Ablauf der Getriebeberechnung

Eine ähnliche Vorgehensweise kann man in der Konstruktion bei der Lösungsfindung beobachten, wobei die Gesamtfunktion eines Produktes zuerst in die sinnvollen Teilfunktionen untergliedert wird, für welche dann erst unabhängig voneinander entsprechende Teillösungen erarbeitet werden können (Feldhusen und Grote, 2013). Außerdem können die beiden genannten Teilprozesse auch als normierte Berechnungen in anderen Prozessen oder sogar eigenständig zum Einsatz kommen.

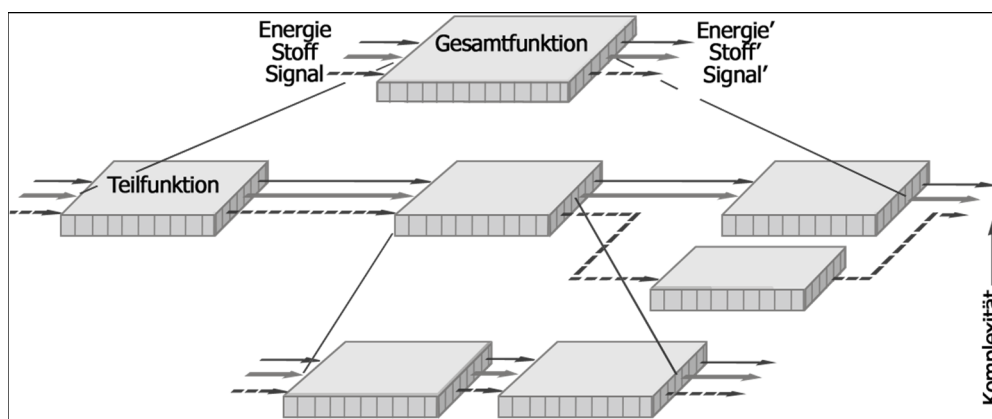


Abbildung 81: Top-Down Prinzip (ebd.)

Teilprozesse müssen nahtlos in einen Prozess eingebunden werden können. D.h. wenn zwei Funktionsbausteine miteinander gekoppelt werden, wird automatisch indem diese auf eine gemeinsame Prozessvariable zugreifen ein Datenfluss hergestellt. Hierbei sind nur die Zugriffsarten unterschiedlich: eine Funktion besitzt einen Schreibzugriff und die andere einen Lesezugriff. Ein Sub-Prozess beinhaltet wie eine Hülle elementare Funktionsbausteine, Kontrollknoten und/oder andere Teilprozesse und hat keinen eigenen Eingabe- und keine Ausgabeparameter. Diese setzen sich aus der

Summe der Eingabe- und Ausgabeparameter der am Teilprozess beteiligten Elemente zusammen. Die Ausführung eines Teilprozesses erfolgt genauso wie die eines Prozesses.

Da die Wiederverwendung vorhandener erfolgreicher Teillösungen ein großes Optimierungspotential im Hinblick auf die Verkürzung von Entwicklungszeiten und ein erfolgreiches Wissensmanagement darstellt, müssen auch hierfür die Voraussetzungen geschaffen werden. Außerdem müssen die Sub-Prozesse, ebenso wie die Funktionsbausteine aus verschiedenen Fachgebieten, schnell wiedergefunden und an die aktuelle Problemstellung adaptiert werden können (Leemhuis, 2005).

Mit einem Sub-Prozess werden mehrere zusammenhängende Prozesselemente beschrieben. Die Darstellung dieser Prozesselemente im Prozess erfolgt dann über ein Symbol des Teilprozesses. Diese Vorgehensweise sorgt für die Erhöhung der Transparenz. An dieser Stelle muss darauf hingewiesen werden, dass es sich bei einem Sub-Prozess um eine rein visuelle Strukturierung (Zusammenfassung) der am Prozess beteiligten Elemente handelt. Ein neuer hierarchischer Aufbau ist damit nicht verbunden. Die Zuordnung der Teilprozesse kann entweder zu einer Klasse oder einer Gruppe erfolgen. Ein benutzerdefinierter Teilprozess kann jederzeit aufgelöst und die darin enthaltenen Prozesselemente können der höheren Ebene des Prozesses zugeordnet werden oder nach einer eventuellen Anpassung einen völlig neuen Teilprozess bilden. Zu den wesentlichen Vorteilen, die sich durch die Nutzung der Subprozesse ergeben, gehört die Möglichkeit komplexe Prozesse übersichtlicher zu gestalten, sowie die gleichen Abfolgen wiederzuverwenden.

### 3.3.3 Fazit

Auf der Basis der im theoretischen Teil der Arbeit durchgeführten Untersuchungen wurde im Hinblick auf die Organisation und die Darstellung der Berechnungsprozesse die Methode „Prozessgebundene Berechnungsbaugruppen“ formuliert. Die Methode basiert auf der formalisierten Darstellung der elementaren Berechnungsaufgaben (Funktionsbausteine) und deren logischen Verknüpfung zu einem Berechnungsprozess in Form eines gerichteten Graphen. Dabei repräsentieren die Knoten dieses Graphen die Funktionsbausteine und die Kontrollflusselemente. Die Kanten des Graphen bilden die logische Abfolge der Prozesselemente ab. Außerdem wurden für die Modellierung der Berechnungsprozesse die minimal notwendigen Modellierungselemente und deren funktionale Eigenschaften identifiziert und beschrieben.

## 3.4 Entwicklung des workfloworientierten Management-Systems

Die informationstechnische Realisierung von dem in der Arbeit aufgestellten Ansatz kann mit Hilfe eines Design Process Management Systems (DPMS) realisiert werden. Workflow Management Systeme unterstützen die Verwaltung (Modellierung, Ausführung und Überwachung) von Workflows. Die WFM Systeme werden üblicherweise bei Prozessen eingesetzt, welche eine Voll- oder zumindest Teilautomatisierung zulassen (Gadatsch, Andreas, o.D.). Im nachfolgenden Kapitel werden die

theoretisch erarbeiteten Ansätze der Methode in Form eines prototypisch entwickelten DPM Systems zur Verwaltung der Berechnungsprozesse vorgestellt. Zu den zwei Hauptzielen des entwickelten IT-Systems gehören die Modellierung und Simulation eines Berechnungsablaufes sowie die Generierung einer ausführbaren Applikation auf Basis des modellierten Ablaufs. Dieses Autorensystem legt einen großen Wert auf die Mensch-Maschine-Schnittstelle und bietet dem Wissensingenieur viele intuitive Funktionalitäten zum Aufbau der Benutzeroberfläche. Ein Schwerpunkt der Programmentwicklung liegt auf der textuellen und graphischen Hilfe sowie auf der Ergebnisinterpretation und der Dokumentationsmöglichkeiten. Der Prototyp nutzt die Ein- und Ausgabeschnittstellen der im Engineering-Umfeld weit verbreiteten Entwicklungsumgebung für Berechnungssoftware MDESIGN. Allerdings wird in dieser Entwicklungsumgebung eine zwar vereinfachte, aber letztendlich doch klassische Programmiersprache für die Beschreibung der Berechnungsroutinen verwendet, deren Terminologie dem Wissensingenieur oft fremd ist (Polyakov und Gründer, 2013).

Das neue IT-Werkzeug soll den Programmieraufwand bei der Beschreibung von Berechnungsroutinen durch den Einsatz vordefinierter Wissensobjekte (Funktionsbausteine) weitgehend vermeiden und die Wiederverwendbarkeit der erfolgreichen oder genormten Teillösungen gewährleisten. Zudem soll durch die graphische Repräsentation der Berechnungen die Transparenz der Lösungen erhöht werden und schließlich soll die Bildung von modular strukturierten Berechnungsbaugruppen möglich sein, die bisher nur als monolithische Programme möglich sind.

### 3.4.1 Design Process Management System (DPMS)

Die Abbildung 82 zeigt einen Überblick über die Architektur des im Rahmen dieser Arbeit entwickelten DPMS. Die wichtigsten Komponenten dieses Systems können der folgenden Auflistung entnommen werden.

- Ein Prozess-Designer für die Modellierung von Berechnungsabläufen,
- Ein Prozess-Interpret für die automatische Umsetzung in ablauffähige Programme,
- Eine Prozess-Engine für die Ausführung der Programme und die Kommunikation mit den internen und externen Ressourcen sowie
- Ein Front-End-GUI für die automatische Bereitstellung von Eingabe- und Ausgabeinformationen sowie die Definition neuer Wissensobjekte.

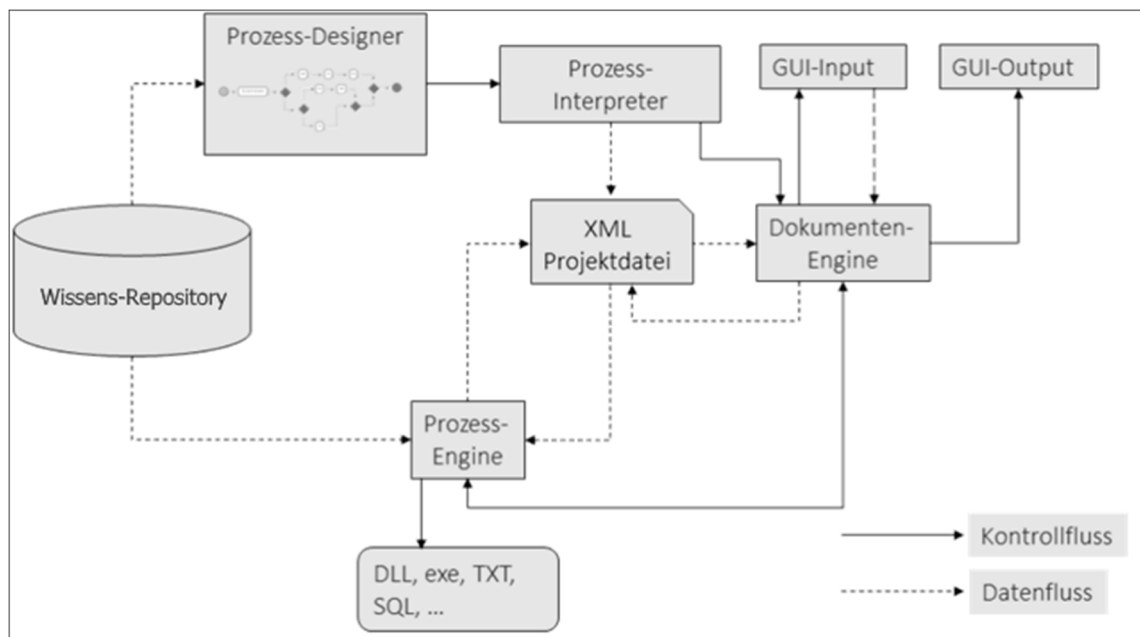


Abbildung 82: Systemarchitektur des Design Process Management Systems (Polyakov und Gründer, 2016, Mashchenko, 2015)

Die Modellierung eines Prozesses erfolgt im Prozess-Designer, welcher die Möglichkeit bietet, die einzelnen Funktionsbausteine und ggfs. Subprozesse graphisch zu einem Gesamtprozess zusammenzusetzen. Der an den Designer angeschlossene Prozess-Interpreter überführt anschließend die graphisch modellierten Zusammenhänge (visuelles Modell) in eine maschinenlesbare Sprache XML. Dieses Beschreibungsformat des Prozesses kann dann von einer Prozess-Engine verstanden und ausgeführt werden. Bei der Übersetzung nach XML handelt es sich um die Konvertierung von Anweisungen und Regeln auf derselben Abstraktionsebene, da die visuelle Sprache bereits eine eindeutige Semantik besitzt. So werden Fehler bei der Konvertierung durch eine eventuell vorhandene semantische Lücke vermieden. Anschließend initialisiert der Interpreter die Dokumenten-Engine, die ihrerseits die Aufgabe hat, eine Eingabemaske zu generieren, die für den Start der Berechnung erforderlich ist. Dies erfolgt automatisch und dient der Eingabe bzw. Sammlung aller für den Start des Prozesses notwendigen Informationen. Die Eingabeparameter werden anschließend in die zentrale Datenbank geschrieben, um sie allen im Prozess beteiligten Funktionsbausteinen und Kontrollflusselementen zugänglich zu machen. Danach wird die Prozess-Engine als die zentrale Komponente des Design Process Management Systems angestoßen. Sie interpretiert die Projekt-XML, koordiniert und synchronisiert entsprechend die einzelnen internen Teilprozesse sowie die externen Applikationen und sorgt so für den Ablauf des kompletten Prozesses. Nach Abschluss des Prozesses liegen alle Ausgabeinformationen in der zentralen Datenbank. Die erneut angestoßene Dokumenten-Engine stellt alle Eingaben und Ergebnisse übersichtlich in Form von Tabellen bzw. Grafiken zusammen. Die Datenbanken sind im Prototyp in Form von Microsoft Access Datenbanken realisiert. Der Umstieg auf die heute dem Stand der Technik entsprechende SQL-Datenbank ist der Weiterentwicklung des Systems vorbehalten.

## Der Prozess-Designer

Die Hauptrolle bei der Generierung der Abläufe übernimmt der sogenannte Prozess-Designer. Seine Aufgabe besteht darin, dem Anwender die Darstellung der gewünschten Abläufe in einer grafischen Sprache zu ermöglichen. Der Arbeitsbereich im Designer wurde in zwei Abschnitte unterteilt. In einem ist das Repository mit den vorhandenen Funktionsbausteinen und den Teilprozessen positioniert, im anderen wird die Modellierung vorgenommen (siehe Abbildung 83).

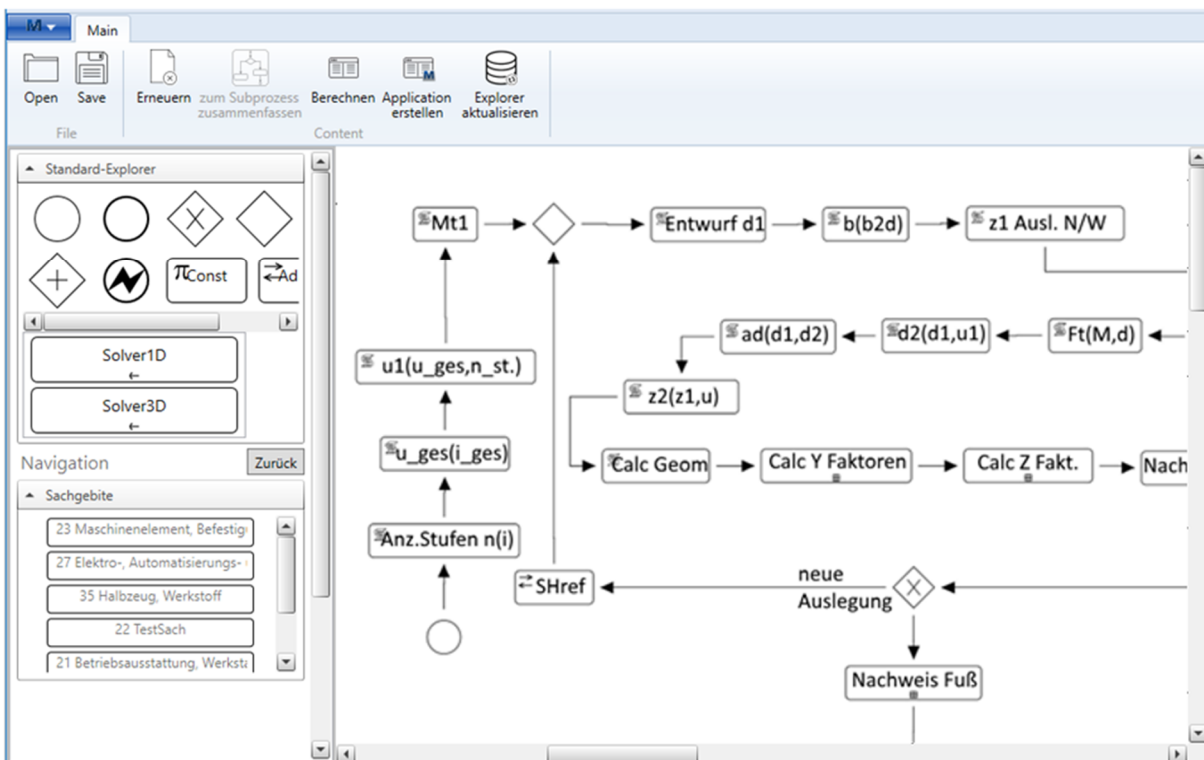


Abbildung 83: GUI vom Prozess-Designer

Die Menüleiste beinhaltet Funktionen zur Erstellung einer neuen Applikation, zum Speichern und Öffnen des modellierten Prozesses und zu dessen Simulation. Das Repository stellt im Standard-Explorer die Steuerelemente für den Ablauf des Prozesses sowie die Funktionsbausteine und Teilprozesse entsprechend der Klassifikation im unteren Bereich des Fensters bereit. Über die Drag & Drop Technik können die Prozesselemente auf die Arbeitsfläche gezogen und dort anschließend vernetzt werden.

Zu den Kontrollknoten des Prozessablaufes gehören obligatorische Elemente, wie zum Beispiel „Start-Knoten“ und „End-Knoten“. Diese kennzeichnen eindeutig den Anfang und das Ende eines Prozesses. Drei zur Verfügung stehende Gateways ermöglichen die Modellierung der Aufspaltung, Zusammenführung sowie das Parallelisieren der Abläufe. Die zwei letzten Elemente (Solver1D und Solver3D) können zur Lösung von inversen Aufgaben besonders für die nichtlinearen Gleichungen und Gleichungssysteme verwendet werden. Alle diese Elemente befinden sich in einem separaten

Standard-Explorer und können vom Benutzer nicht angelegt oder geändert werden. Im Gegensatz dazu ist der fachliche Teil des Repository erweiterbar. Entsprechend der Klassifizierungsregeln nach eCl@ss können Funktionsbausteine und Teilprozesse dort neu angelegt werden. Weitere Informationen zum Anlegen eines neuen Merkmals, eines neuen Funktionsbausteins oder eines Teilprozesses sind in dem Kapitel 3.4 näher beschrieben.

Die Vernetzung der Funktionsbausteine untereinander sowie mit den Kontrollflusselementen erfolgt über die Verbindung der Koppelstellen bzw. auch Ports genannt (siehe Abbildung 84).

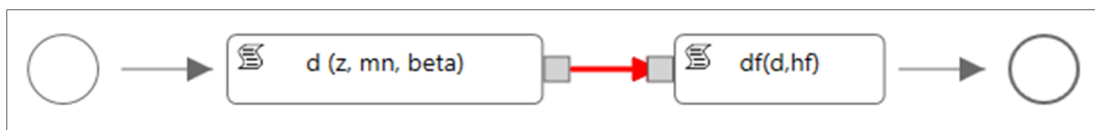


Abbildung 84: Verbindung zweier Funktionsbausteine

Diese Ports werden eingeblendet, sobald der Mauszeiger sie berührt. Analog funktioniert dies auch mit den Kontrollelementen. Durch eine solche Verbindung wird die Reihenfolge der Ausführung der Prozessbeteiligten definiert. Die Verbindungsrichtung spielt dabei keine Rolle. Die Prozesselemente können frei von links nach rechts oder von unten nach oben verbunden werden. Entscheidend ist nur, auf welches Objekt die Spitze der Kante zeigt. Dieses wird dann als nächstes abgearbeitet.

Eine weitere wichtige Voraussetzung für die Modellierung ist die Verfügbarkeit der an einem Funktionsbaustein beteiligten Merkmale. Wird auf die Arbeitsfläche ein Funktionsbaustein gezogen, dessen Merkmalstypen in der Berechnung noch nicht vorhanden sind, müssen alle für ihn notwendigen Input- und Output - Merkmale erzeugt werden.

Die Bezeichnungen und Symbole können im Eigenschaftsfenster eines Funktionsbausteines frei editiert werden (siehe Abbildung 85). Dieses Dialogfenster kann über einen Mausklick auf den Funktionsbaustein aufgerufen werden. Dabei beinhaltet die letzte Zeile der Tabelle immer das Ausgabemerkmal (in diesem Fall den Teilkreisdurchmesser eines Zahnrades), alle restlichen sind Eingabeparameter.

**Eigenschaften: d (z, mn, beta)**

Name	Bezeichnung	Symbol
R_z ▼	Zähnezahl (Ritzel)	$z^{s1}$
R_mn ▼	Normalmodul	$m^{sn}$
R_beta ▼	Schrägungswinkel	$\alpha^{s2}$
R_d ▼	Teilkreisdurchmesser (Ritzel)	d

Abbildung 85: Input- und Output-Eigenschaften eines Funktionsbausteines

Sind beim Hinzufügen eines neuen Funktionsbausteins ein oder mehrere Merkmale mit den, für den Funktionsbaustein notwendigen Typen im Prozess vorhanden, wird der Anwender darüber mit Hilfe des entsprechenden Hinweises informiert. Die Entscheidung, ob ein neues Merkmal erforderlich ist oder eines aus der Liste der vorhandenen Merkmale genügt, ist vom Berechnungsablauf abhängig. Wenn zum Beispiel ein Baustein für die Berechnung des Teilkreisdurchmessers hinzugefügt wird, bedeutet es meistens, dass die Berechnung für ein anderes Rad (z. B. das Gegenrad) erfolgen soll. In diesem Fall kann über das Eigenschaftsfenster (siehe Abbildung 85) in der Spalte „Name“ ein neues Merkmal dieses Typs unter der Angabe des Namens, der Bezeichnung und des Symbols erstellt werden (siehe Abbildung 86).

**Neue Variable hinzufügen: d (z, mn, beta)**

Name	Bezeichnung	Symbol
R_d2	Teilkreisdurchmesser (Rad)	$d^{s2}$

Abbildung 86: Definition eines neuen Merkmales für einen Funktionsbaustein

Weitere Funktionen des Prozess-Designers sind die Definition eines neuen Merkmals und/oder Funktionsbausteins sowie das Erstellen und Ablegen neuer Teilprozesse in dem Wissens-Repository.

Falls bei der Definition eines neuen Funktionsbausteins das benötigte Merkmal nicht im Repository vorhanden ist, kann unter der Berücksichtigung der eCl@ss Vorschriften zur Definition neuer Merkmale dieses neu angelegt werden. Für die Eingabe der Merkmale wurden im Rahmen der vorliegenden Arbeit entsprechende Werkzeuge und Schnittstellen entwickelt, welche das Füllen der Merkmalsdatenbank erlauben. Das Prinzip der Merkmalseingabe ist in der Abbildung 87 dargestellt.

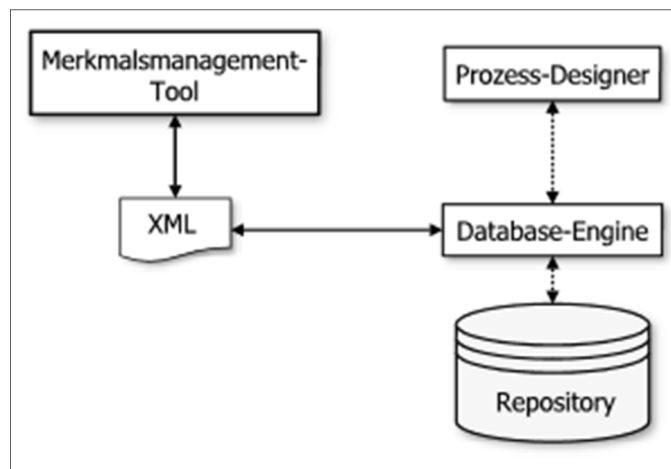


Abbildung 87: Schema für die Definition eines neuen Merkmals

Das Anlegen des Merkmals erfolgt mit Hilfe eines Merkmalsmanagement-Tools (hierfür wird MDESIGN author verwendet). Nach dem Aufruf der entsprechenden Funktion im Prozess-Designer wird mittels der Database-Engine die komplette Liste der Merkmale der gewünschten Klasse aus dem Repository geladen und in eine XML-Datei geschrieben. Diese XML-Datei wird dann automatisch mit dem Merkmalsmanagement-Tool geöffnet und dargestellt (siehe Abbildung 88).

Ersatzzahnanzahl		$z_n =$	0
Normalmodul		$m_n =$	0 mm
Stirnmodul		$m_t =$	0 mm
Schrägungswinkel		$\beta =$	0 °
Normaleingriffswinkel		$\alpha_n =$	0 °
Grundschrägungswinkel		$\beta_b =$	0 °
Zähnezahl		$z =$	0
Zahnbreite		$b =$	0 mm
Profilverschiebungsfaktor		$x =$	0
Teilkreisdurchmesser		$d =$	0 mm
Grundkreisdurchmesser		$d_b =$	0 mm
Betriebswälzkreisdurchmesser		$d_w =$	0 mm
Fußkreisdurchmesser		$d_f =$	0 mm
V-Kreis-Durchmesser		$d_v =$	0 mm
Kopfkreisdurchmesser		$d_a =$	0 mm

Abbildung 88: Auszug aus der Merkmalsleiste der Klasse Stirnrad



Alle diese Merkmale können dabei bei Bedarf editiert werden. Für das Anlegen eines neuen Merkmals müssen alle seine minimal notwendigen Eigenschaften wie Name, Bezeichnung, Symbol, Quelle etc. angegeben werden. Anschließend werden die Änderungen in die zuvor geladene XML-Datei geschrieben und wieder an die Database-Engine übergeben, welche diese Datei interpretiert und entsprechenden Änderungen im Repository vornimmt.

Die Definition eines neuen Funktionsbausteins erfolgt analog und kann auch über den Prozess-Editor während der Modellierung erledigt werden. Die Informationen zu dem neuen Funktionsbaustein werden aber nicht in eine Schnittstellendatei, sondern direkt in das Repository geschrieben. Nähere Informationen zur Definition eines Funktionsbausteins sind in Kapitel 3.4.3 zu finden.

Soll ein Ablauf wie zum Beispiel die Geometrieberechnung des Stirnradpaares nach DIN 3960 / DIN ISO 21771 als Teilprozess wiederverwendet werden, kann dieser als „Master-Prozess“ bzw. Sub-Prozess im Wissens-Repository abgelegt werden. Oft ist dies allein aus der Sicht der Zeitersparnis, welche durch die Wiederverwendung des gleichen Berechnungsablaufes mit anderen Parametern gewonnen werden kann, sinnvoll. Ein Beispiel sind mehrstufige Stirnradgetriebe, bei denen die Bestimmung der Geometrie je Stufe immer nach dem gleichen Schema abläuft. Der Teilprozess für die Berechnung der Geometrie des Stirnradpaares kann in der Klasse des Stirnratsatzes 23-17-01-10 platziert werden. Im Gegensatz zum Anlegen eines neuen Funktionsbausteins, dessen Zuordnung nur zu einer Klasse möglich ist, welche eindeutig durch den Typ des Ergebnismerkmals vorbestimmt ist, können Teilprozesse in der eCl@ss-Hierarchie auch an die Gruppen angehängt werden.

### 3.4.2 Datenschnittstellen – Projekt-Datei

Alle Kommunikationsdateien zwischen der Dokumenten-Engine und dem Prozess-Designer sowie der Prozess-Engine sind im XML Format aufgebaut. Dieses vom World Wide Web Consortium-W3C (W3C, 2006) spezifizierte und empfohlene Format für maschinenlesbare Dokumente stellt heute den Standard für den elektronischen Datenaustausch dar. XML-Dokumente sind relativ einfach und klar strukturiert. Wie alle Markup-Sprachen in der Informatik dient auch XML dazu, bestimmte Teile der Dokumente mit zusätzlichen Informationen (Metadaten) zu versehen und damit maschinenlesbar (verarbeitbar) zu machen. Das Annotieren der Dokumententeile erfolgt mit den sogenannten „Tags“ (dt. Etikett oder Schild). Bis zu einem bestimmten Grad können die XML-Dokumente auch von Menschen gelesen und interpretiert werden. Der allgemeine Aufbau eines XML-Dokuments ist in der Abbildung 89 dargestellt.

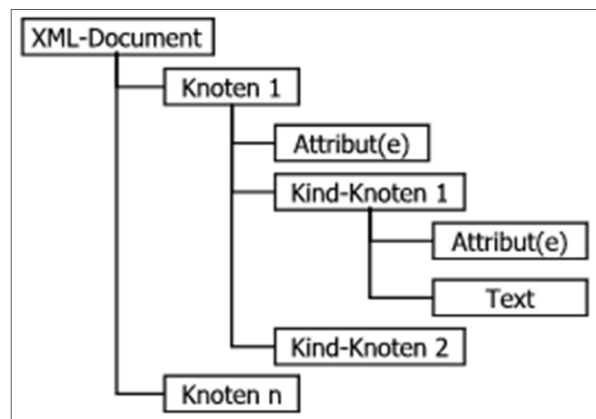
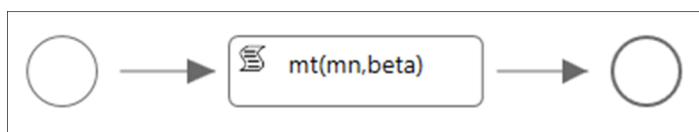


Abbildung 89: Struktur eines XML-Dokumentes

Wie zu erkennen ist, verwendet XML als Datenmodell eine Baumstruktur. Dabei entspricht jeder Tag einem Knoten mit einem Namen im Datenmodell und jeder verschachtelte Tag einem Kind-Knoten. Im Gegensatz zu einem einfachen Textdokument ist ein XML-Dokument an ein bestimmtes Schema gebunden und kann gegen dieses validiert werden. Diese Struktur bildet die Basis für den weiterreichenden Einsatz von XML-Dokumenten in verschiedenen Anwendungsumgebungen, in dem sich die Partner an eine vereinbarte schematische Definition von Dokumenten und Daten gebunden haben (Hitzler, et al., 2007).

#### Projekt-Datei

Eine Schlüsselrolle spielt in dem entwickelten Design Process Management System die sogenannte Projekt-Datei. Diese Datei beinhaltet die komplette Beschreibung des modellierten Berechnungsablaufs und dient der Koordination der Ausführung durch die Prozess-Engine. In der Abbildung 90 ist ein einfaches Beispiel einer Berechnung graphisch dargestellt.

Abbildung 90: Graphische Darstellung der Berechnung des Stirnradmoduls  $m_t$ 

Dieses Beispiel beinhaltet nur die minimal notwendigen Modellierungselemente zur Ausführung der genannten Berechnung: Start-Knoten, End-Knoten und den eigentlichen Funktionsbaustein zur Berechnung des Stirnmoduls  $m_t$  nach DIN 3960 (siehe Gleichung (3-5)). Die Projekt-Datei dieser Berechnung ist im Anhang B zu finden. Das erste Element in der Liste ist der Start-Knoten mit der ID=20528282. Diese achtstellige Nummer wird mit Hilfe eines Random-Generators für alle Prozesselemente und Prozessvariablen erzeugt und sorgt für eine eindeutige Identifikation der Prozessbeteiligten unabhängig von deren Anzahl in einem Prozess. Die Aufgabe eines Start-Knotens ist die Definition des Anfangs eines Berechnungsablaufs sowie mit Hilfe einer Connection der Verweis auf

das nächste Element in dem Ablauf. Außerdem können bei dem Start-Knoten genauso wie bei den anderen Funktionsbausteinen und Kontrollflusselementen die Eingabe-Parameter angegeben sein. Die explizite Angabe der Merkmale an einem Start-Knoten kann zu einer frühzeitigen Kennzeichnung der im Prozess vorhandenen Eingangsparameter bzw. bekannten Randbedingungen genutzt werden. In diesem Fall erfüllt der Start-Knoten quasi die Funktion eines Containers für die Sammlung der Eingabeparameter und gibt diese anschließend direkt zur Eingabe frei. In dem oben aufgeführten Beispiel erfolgt die Angabe der Eingangsparameter aber implizit über den Funktionsbaustein zur Berechnung der gesuchten Größe  $m_t$ , da es nur wenige Parameter sind. Ausgabeparameter sind bei dem Start-Knoten nicht zulässig, da dieses Element keine Verarbeitung der Daten durchführen kann. Ein sehr ähnliches Konstrukt weist der End-Knoten auf. Dieser kennzeichnet das Ende eines Ablaufs und kann die Merkmale explizit für die Ausgabe beinhalten. Der Funktionsbaustein  $m_t = f(m_n, \beta)$  beinhaltet zwei Eingangsparameter (Normalmodul  $m_n$  und Schrägungswinkel  $\beta$ ), welche über die entsprechenden IDs verlinkt sind. Als Ausgabeparameter ist das gesuchte Merkmal  $m_t$  angegeben. Über die eindeutige Identifikation des Typs des Funktionsbausteins (Func="P4cceLVW40eP2") wird seine Funktion zum Zeitpunkt der Ausführung der Wissensdatenbank entnommen und mit den entsprechenden Werten der Eingangsparameter ausgerechnet. Alle Prozessvariablen sind in der Projekt-Datei in dem Bereich „Ports“ aufgelistet. Zu den erforderlichen Eigenschaften der Parameter gehören deren Typ, ID und Angabe der Prozessknoten, an denen sie als Eingabe- bzw. Ausgabeparameter beteiligt sind. Der letzte Block der Projekt-Datei (Connections) beschreibt die logische ablaforientierte Verknüpfung der Prozessknoten untereinander.

### 3.4.3 Definition eines Funktionsbausteines

Nachfolgend wird ein Beispiel für die Definition eines Funktionsbausteins aus der Geometrieberechnung einer Verzahnung am Beispiel der Ermittlung des Stirnmoduls  $m_t$  nach DIN 3960:1987-03 erläutert. Die Berechnungsgleichung lautet:

$$m_t = \frac{m_n}{\cos(\beta)} \quad (3-5)$$

Wobei  $m_n$  – Normalmodul und  $\beta$  – Schrägungswinkel sind. Zunächst muss gewährleistet werden, dass alle drei Größen als Merkmale in der Wissensdatenbank vorhanden sind. Diese Größen beschreiben alle ein Zahnrad. Ein Zahnrad mit dem bestimmten Modul und dem Schrägungswinkel kann vorerst ohne Rücksicht auf andere künftige Getriebeeigenschaften eigenständig existieren. Erst bei der Paarung des Rades mit einem Gegenrad kommen zusätzliche aufgabenspezifische Randbedingungen (z. B. ob die Schrägungswinkel beider Räder gleich sein müssen oder ein Achsversatz gewährleistet werden soll etc.) ins Spiel. Diese werden dann auf der Ebene des Zahnradpaares betrachtet. Falls die gewünschten Merkmale nicht in der Wissensdatenbank vorhanden sind, können diese mit Hilfe des

Prozess-Designers angelegt werden. Als nächstes wird der Funktionsbaustein selbst definiert, ebenso mit Hilfe des Prozess-Designers (siehe Abbildung 91)

Abbildung 91: Anlegen eines Funktionsbausteines

Dabei müssen die Eingabeparameter, das Ergebnismerkmal und das auszuführende Skript der mathematischen Formel oder eines Algorithmus angegeben werden.

Im Bereich Eingabeparameter können die Eingabegrößen (Eingangsmerkmale) mittels Auswahl aus der Liste der vordefinierten Merkmale entsprechend der Klassifizierung definiert werden. Außerdem müssen zu dem neuen Funktionsbaustein auch die Hilfeinformationen ausgefüllt werden.

#### Hilfedefinition eines Funktionsbausteines

Zur Eingabe und Verwaltung der Hilfe-Informationen über einen Funktionsbaustein wurde eine Maske mit den entsprechenden Eingabefeldern implementiert (siehe Abbildung 92, links). Alle Felder sind für mehrere Sprachen verfügbar. Mindestens eine Sprache muss komplett ausgefüllt sein. Alle Angaben sind verpflichtend. Eine Ausnahme bildet der Berechnungshintergrund. Dieser kann entweder als Text oder in Form einer Graphik angegeben werden. Nach der Eingabe der sogenannten Metadaten werden sie in einer Datenbanktabelle inklusive der Angabe der Identifikationsnummer des Funktionsbausteins abgelegt. Diese abgespeicherten Informationen werden beim Aufruf des Hilfesystems eines Funktionsbausteins automatisch in Form einer HTML Seite zusammengefasst (siehe Abbildung 92, rechts). Ein entscheidender Vorteil dabei ist die Einheitlichkeit, denn die Informationen werden für alle Funktionsbausteine in gleichem Format und in gleicher Reihenfolge angezeigt.

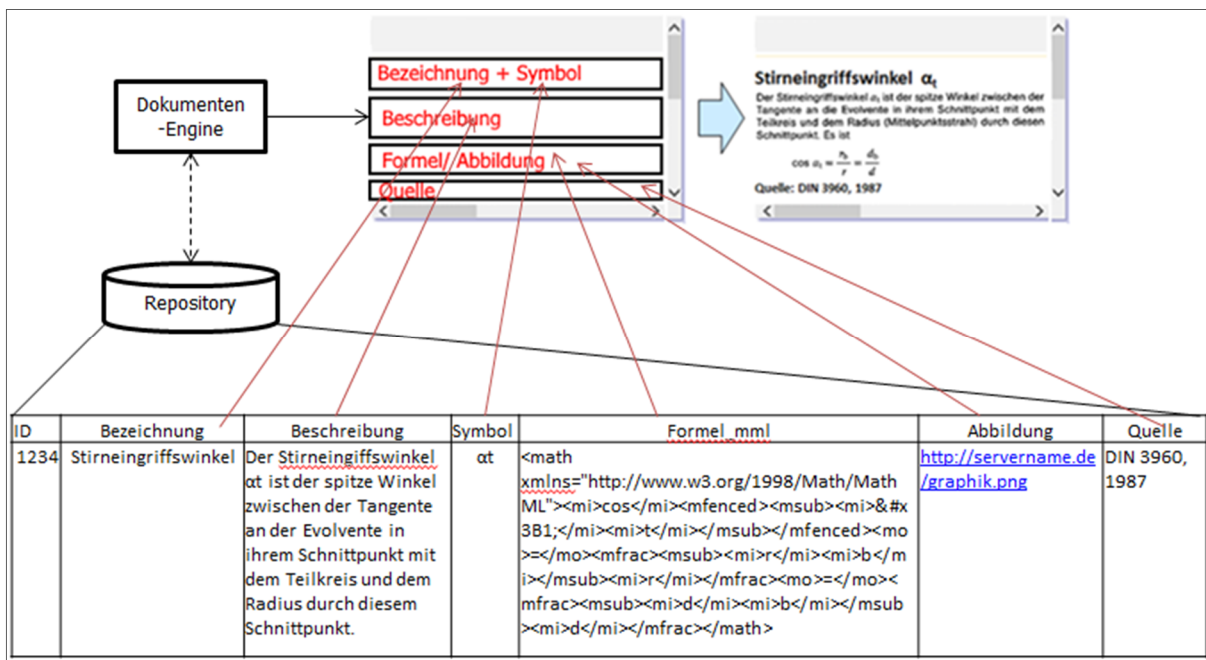


Abbildung 92: Aufbau des Infosystems

Die Summe aller Informationen der an einem Prozess beteiligten Funktionsbausteine ergibt zusammen mit den entsprechenden Merkmalshilfen die Berechnungsdokumentation eines Ablaufs.

### 3.4.4 Modellierung von Berechnungsprozessen

Generell verlaufen Prozesse nur selten sequentiell, die meisten weisen dagegen zahlreiche Verzweigungen und Rückkopplungen auf. Sie können mit Hilfe von Gateways koordiniert werden und Kontrollflüsse auseinander- oder zusammenfließen lassen. Gateways gibt es in verschiedenen Ausprägungen, wie zum Beispiel exklusive, inklusive oder parallele Gateways. Zur Veranschaulichung des Sachverhalts wird nachfolgend am Beispiel der Geometrieberechnung der Verzahnung die Notwendigkeit der Anwendung von Kontrollelementen wie Gateways dargestellt. Als Beispiel dient der in der Abbildung 93 dargestellte Berechnungsablauf.

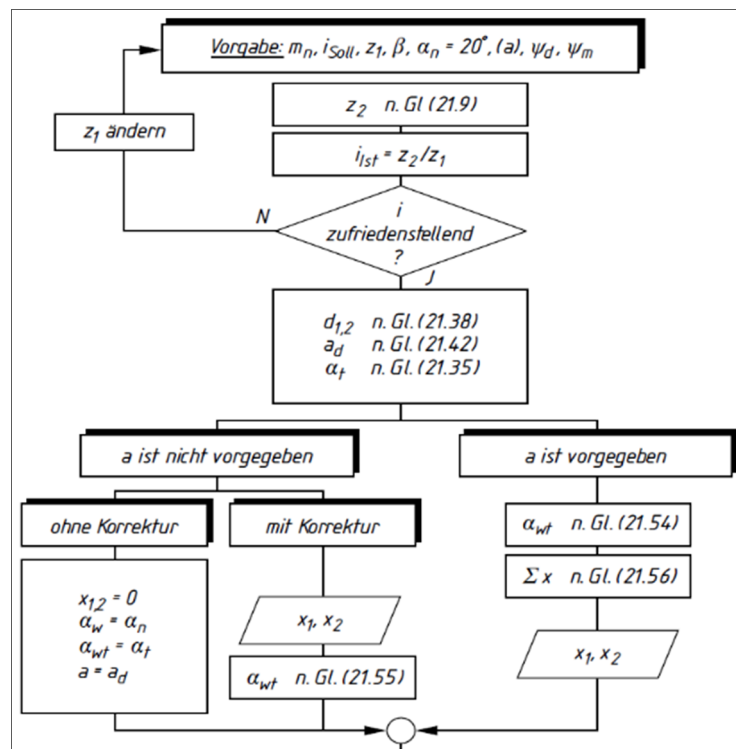


Abbildung 93: Auszug aus der Geometrieberechnung der Verzahnung (Wittel, et al., 2011)

Der hier abgebildete Berechnungsverlauf widerspricht zwar nicht den Vorgaben in den Verzahnungsnormen, jedoch sind die verwendeten Berechnungsgleichungen etwas allgemeiner gehalten als in den Normen.

Für die anschließende Automatisierung des Berechnungsablaufes ist in diesem Beispiel besonders die Unterscheidung nach dem Achsabstand und seiner Korrektur (Profilverschiebungsfaktoren) interessant. Diese Größen (Achsabstand und Profilverschiebungsfaktoren) sind zusammenhängend und können nicht alle auf einmal vorgegeben werden, da sonst die Aufgabe überbestimmt ist. Aus diesem Grund muss im Laufe des Berechnungsprozesses unterschieden werden können, welche von den Größen bestimmt werden sollen und welche schon vorhanden sind. Dieser Sachverhalt kann mit Hilfe eines exklusiven Gateways realisiert werden, welches eine logische XOR-Semantik umsetzt.

Die Umsetzung des Beispielprozesses ist in der

Abbildung 94 dargestellt. Nach der Berechnung des Stirneingriffswinkels  $\alpha_t$  wird unterschieden, ob der Wert des Achsabstandes  $a$  vorgegeben wurde, denn daraus folgt die Wahl des Funktionsbausteins zur Berechnung des Betriebseingriffswinkels  $\alpha_{wt}$ .

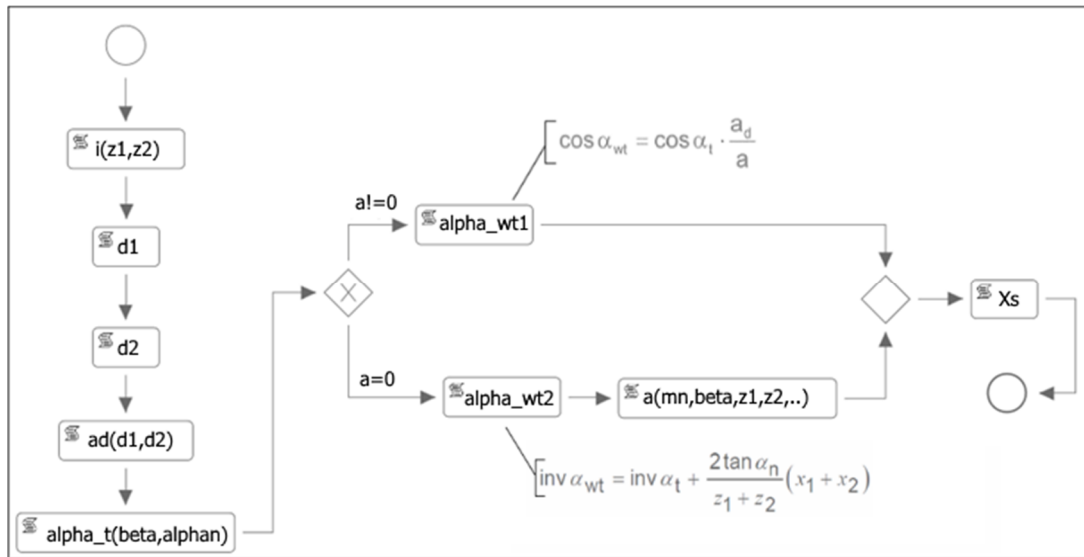


Abbildung 94: Umsetzung des Beispielprozesses

Zur Definition des Gateways gehört die Angabe der Bedingung, in deren Abhängigkeit der eine oder der andere Zweig der Berechnung weitergeführt werden soll (siehe Abbildung 95).

Abbildung 95: Definition der XOR-Auswahl

Zusätzlich zum Skript für die Ausführung werden das Element und sein Merkmal (Steuermerkmal) ausgewählt, dessen Wert für die Bedingung relevant ist. Abhängig von der Bedingung können mehrere Merkmale aus verschiedenen Elementen verwendet werden. Optional können zusätzlich die Kanten nach dem Gateway beschriftet werden. Dies wird auch empfohlen, denn dadurch wird die Lesbarkeit erhöht. Nicht zu vergessen ist die Tatsache, dass das XOR –Gateway in diesem Beispiel geschlossen werden soll. Dies bedeutet, dass noch ein XOR-Gateway vor der Berechnung der Summe der Profilverschiebungsfaktoren platziert werden muss. Die Abarbeitung des letzten Funktionsbausteins beginnt erst, wenn einer der beiden Zweige das zweite Gateway erreicht hat.

Eine weitere Gateway-Spezialisierung stellt das Parallele Gateway dar. Ein Parallel-Gateway benötigt keine zusätzlichen Bedingungen, denn es lässt die Ausführung aller ausgehenden Prozesszweige gleichzeitig und unabhängig voneinander zu. Die Ausführung eines Prozesses nach einem zusammenfließenden Parallel-Gateway wird erst fortgesetzt, nachdem alle Berechnungszweige beendet wurden und an dem zusammenführenden Ende angekommen sind. Besonders hilfreich kann ein Parallel-Gateway bei der mehrfachen Verwendung von komplexen Sub-Prozessen sein, denn es hilft den Berechnungsprozess entsprechend zu beschleunigen und gleichzeitig die Prozesslogik zu verdeutlichen. In der Abbildung 96 werden also nach der Ausführung des Sub-Prozesses zur Prüfung der Eingabedaten sowohl die Berechnung der Geometrie vom Ritzel als auch vom Rad ausgeführt.

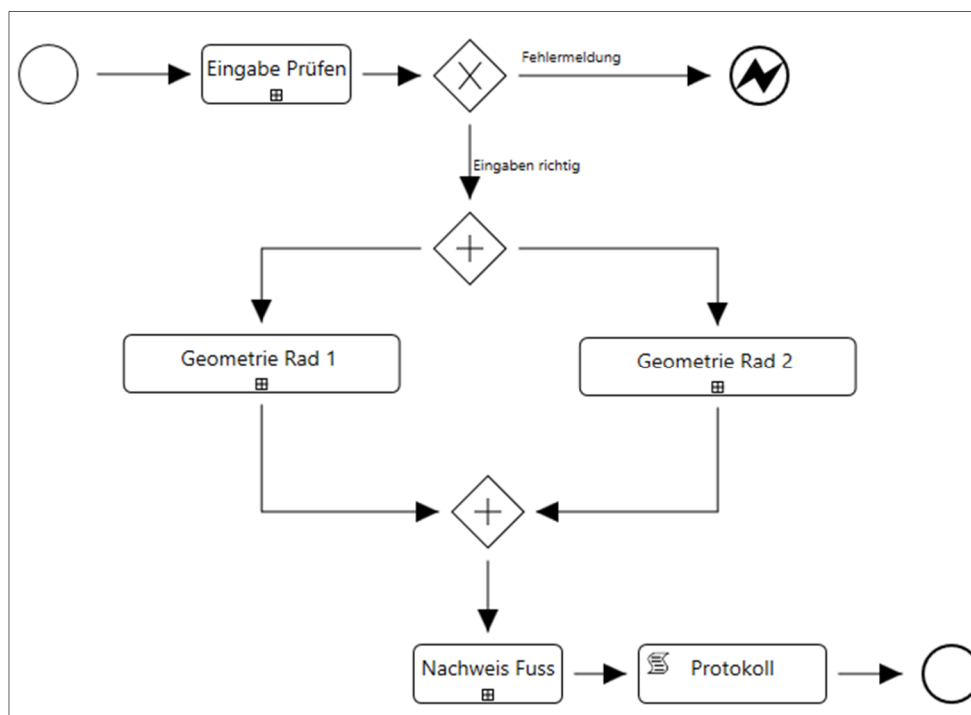


Abbildung 96: Beispielprozess mit Parallel-Gateways.

Erst nach Beendigung dieser beiden Sub-Prozesse wird mit der Berechnung des Tragfähigkeitsnachweises fortgefahren (Götz, 2011).

Obwohl ein Gateway als Raute dargestellt wird, können in der Summe mehr als vier ausgehende sowie eingehende Ablaufflüsse verknüpft werden. Die Steuerung des Ablaufs mit den Gateways erfordert eine implizite Definition der zusätzlichen Modellelemente. In vielen Sprachen (BPMN, UML, etc...) existieren dazu entsprechende Alternativen, wie zum Beispiel die „Bedingte Kante“. Mit Hilfe solcher Kanten kann der Ablauf ähnlich wie mit den Gateways beeinflusst werden. In der Abbildung 97 ist ein Beispielprozess für die Ermittlung der Knicklänge eines Stabes nach Euler aufgeführt.



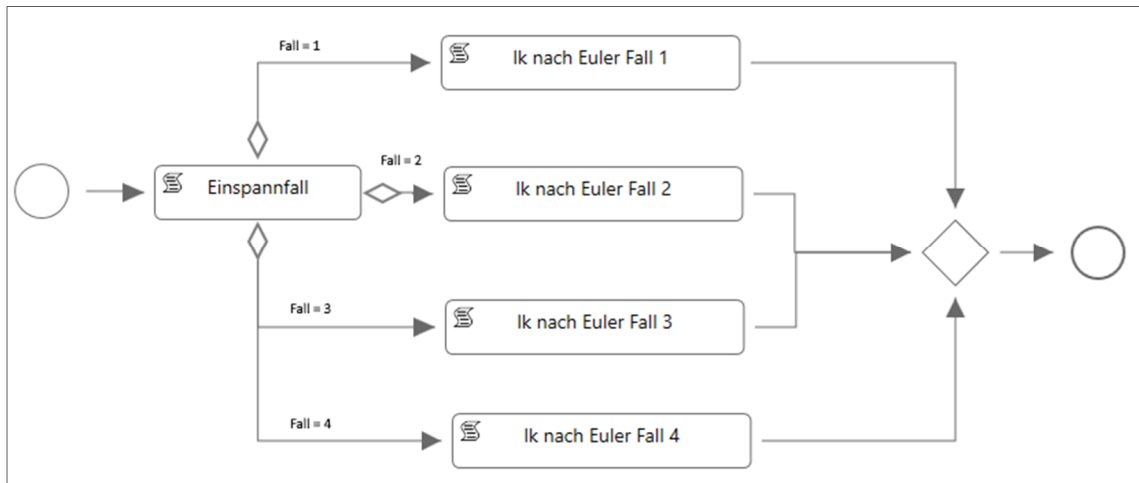


Abbildung 97: Beispielprozess mit bedingten Kanten

Der Vorteil der Bedingten Kante liegt hauptsächlich darin, dass diese direkt von einem Funktionsbaustein ausgehen kann. Die Definition der Bedingung erfolgt dabei genauso wie bei dem Gateway. Im Prinzip sind beide Modellierungsansätze gleichwertig. Welcher angewendet wird, liegt in der bevorzugten Entscheidung des Modellierers.

Wird auf die Ausmodellierung der Abläufe mit Gateways verzichtet, können sehr schnell Abläufe mit den Endlosschleifen entstehen. Dies ist ein sehr häufiger Fehler beim Modellieren von Rückkopplungen und Verzweigungen. Ein einfaches Beispiel einer solchen Endlosschleife ist auf der Abbildung 98 zu sehen.

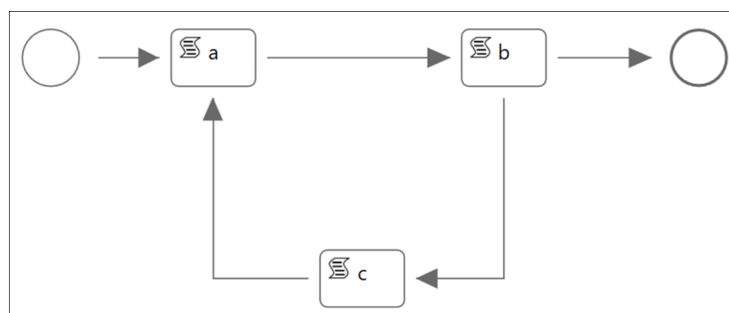


Abbildung 98: Beispiel einer Endlosschleife

Zur Vorbeugung solcher Fehler kann die Zyklensuche genutzt werden. Diese kann mit Hilfe der einfachen Algorithmen aus der Theorie der gerichteten Graphen realisiert werden. Der Berechnungsablauf entspricht im Grunde genommen schon einem gerichteten Graphen mit einem definierten Start- und End-Knoten. In Abbildung 99 ist ein Beispiel eines einfachen Algorithmus für die Identifizierung der Endlosschleifen dargestellt.

```

1  Procedure ZYKLENSUCHE (Knoten x)
2    If Markierung (x) = „in Bearbeitung“ then Zyklus gefunden
3    Else if Markierung(x) = „noch nicht begonnen“ then
4      Markierung(x)= „in Bearbeitung“;
5    For all Nachfolgerknoten y von x do ZYKLENSUCHE(y);
6    Markierung(x)= „abgeschlossen“

```

Abbildung 99: Algorithmus für die Zyklensuche (Schlingloff, 2006)

Mit dem angegebenen Algorithmus kann festgestellt werden, ob von einem bestimmten Knoten ein Zyklus zu erreichen ist. Im Prinzip wird bei dem ersten Knoten angefangen und entlang der Kanten des Graphen fortgeführt. Dabei werden die Knoten, bis ihre Bearbeitung abgeschlossen ist, mit einem Marker „in Bearbeitung“ versehen. Wird ein Knoten, der den Marker „in Bearbeitung“ besitzt wieder erreicht, bedeutet dies automatisch, dass ein Zyklus erreicht wurde. Diese Lösung ist ausreichend, um dem Anwender den Hinweis zu geben, dass ein bestimmter Funktionsbaustein sich in einer Endlosschleife befinden könnte.

### Lösung von inversen Aufgaben

Bei vielen Aufgabenstellungen in der Konstruktion sind die Zielgrößen zwar bekannt, die Randbedingungen bzw. die Eingangsgrößen müssen jedoch erst ermittelt werden. Häufig fehlt es dabei an Möglichkeiten oder an der Zeit für eine explizite Ausarbeitung und das Herleiten der Zusammenhänge aus bekannten Sachverhalten. Oft ist bei einem bekannten Funktionswert auch eine schnelle Berechnung des gewünschten Parameters gefragt. Solche Aufgaben zählen zu den inversen und können bei Bedarf am schnellsten mit Hilfe numerischer Methoden gelöst werden. Aus diesem Grund wurde im Rahmen der prototypischen Implementierung des Ansatzes für Prozessgebundene Berechnungsbaugruppen das Newton-Verfahren implementiert. Da im Rahmen der Entwicklung des Autorensystems ein Less-Code Ansatz angestrebt wird, ist es sinnvoll, solche Elemente, wie numerische Verfahren zum Lösen von nichtlinearen Gleichungen als Standard-Modellierungselement zu haben. Bei PTC Mathcad, einem Gleichungslöser der Firma Parametric Technology Corporation in USA, sind beispielsweise die Schlüsselwörter „Vorgabe“ und „Lösen“ vorhanden, welche die Möglichkeit bieten, auch ohne tiefe Mathematikkenntnisse die Lösung inverser Aufgaben sowie der nichtlinearer Gleichungen zu bewältigen.

Mit Hilfe der Gleichung (3-6) für die Bestimmung des Achsabstandes nach DIN ISO 21771:2014-08 kann die Lösung der inversen Aufgabe vereinfacht veranschaulicht werden.

$$a = \frac{|z_1 + z_2| \cdot m_n \cdot \cos(\alpha_t)}{2 \cdot \cos(\alpha_{wt}) \cdot \cos(\beta)} \quad (3-6)$$

Dabei müssen Größen wie der Normalmodul  $m_n$ , die Anzahl der Zähne vom Ritzel und Rad  $z_{1,2}$ , der Schrägungswinkel  $\beta$ , der Stirneingriffswinkel  $\alpha_t$  sowie der Betriebseingriffswinkel  $\alpha_{wt}$  bekannt sein.

Die Bestimmung des Betriebseingriffswinkels  $\alpha_{wt}$  beim bekannten Achsabstand  $a$  erfordert die Umstellung dieser Gleichung nach  $\alpha_{wt}$  und das Anlegen eines neuen Funktionsbausteins. Der Aufwand der Erstellung eines solchen Funktionsbausteins ist in diesem Fall nicht groß, kann aber abhängig von der Komplexität des Zusammenhangs entsprechend steigen. Die Wiederverwendung von Funktionsbausteinen zur Bestimmung deren Parametern bei bekanntem Funktionswert kann durch die Anwendung des Solver1D-Bausteins eine erhebliche Zeitersparnis beim Modellieren bringen. Den bereits vorhandene Funktionsbaustein für die Berechnung des Achsabstandes  $a$  kann wieder genutzt werden, um dieses Mal den Funktionsparameter  $\alpha_{wt}$  zu ermitteln.

Dazu muss lediglich die zu ermittelnde Prozessvariable mit dem entsprechenden Startwert angegeben werden (siehe Abbildung 100). Bereits nach wenigen Iterationen ist der gesuchte Wert ermittelt, ohne dass der Ingenieur sich mit den zentralen Differenzenquotienten oder den Operationen mit Jacobi-Matrizen beschäftigen muss.



Abbildung 100: Eigenschaften des Solver1D-Bausteines

Dadurch können die aus der Nachrechnung bekannten Zusammenhänge direkt als Auslegungsroutinen fungieren, ohne die komplex ausgefallenen Umstellungen durchführen zu müssen. Vorausgesetzt ist natürlich, dass die stetige Differenzierbarkeit der Nachrechnungsfunktionen gegeben ist. Da bei der Entwicklung des Prozess-Designers der Less-Code Ansatz angestrebt wird, ist es sinnvoll, den Umgang mit den inversen Berechnungen mit Hilfe entsprechender vorprogrammierter Werkzeuge (Features) zu unterstützen, anstatt gleich eine neue Funktionsprogrammierung vorzunehmen.

### 3.4.5 Ausführung von Prozessen und Sub-Prozessen

Eines der Ziele der vorliegenden Arbeit ist die automatische Ausführung eines Berechnungsmodells. Die Ausführung des Prozesses untergliedert sich in drei Hauptbestandteile. Zu Beginn müssen die für die Ausführung eines Prozesses erforderlichen Parameter mit den entsprechenden Werten belegt

werden. Dazu wird eine sogenannte Eingabeseite dynamisch generiert. Die Eingabeparameter eines Prozesses ergeben sich automatisch aus allen freien Parametern der beteiligten Funktionsbausteine. In der Abbildung 101 ist ein einfacher Beispielprozess dargestellt, in dem auch die freien und die zu übergebenden Parameter ersichtlich sind.

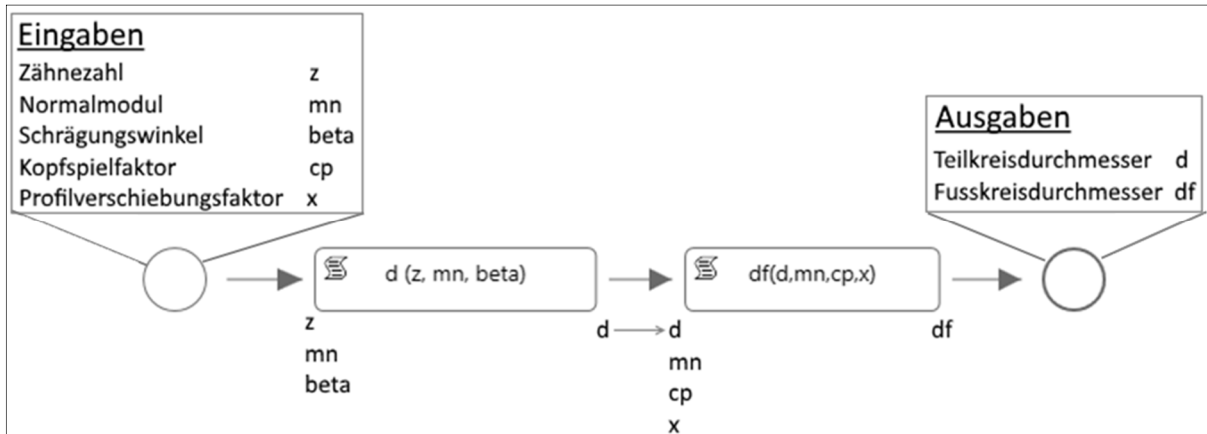


Abbildung 101: Beispiel eines Teilprozesses

Die Eingangsparameter Zähnezahl  $z$ , Normalmodul  $m_n$  und Schrägungswinkel  $\beta$  des ersten Funktionsbausteins zur Berechnung des Teilkreisdurchmessers, genauso wie die Parameter Kopfspielfaktor  $c_p$  und Profilverschiebungsfaktor  $x$  sind freie Eingangsparameter, da diese von keinem Funktionsbaustein berechnet werden. Diese Parameter werden alle zur Belegung über die Eingabeseite freigegeben. Im Gegensatz dazu wird der Eingangsparameter Teilkreisdurchmesser  $d$  des zweiten Funktionsbausteins vom ersten Funktionsbaustein ausgerechnet. Zusammen mit dem Fußkreisdurchmesser  $d_f$  wird er nach dem Ausrechnen auf der Ausgabeseite (siehe Abbildung 102 und Abbildung 103) abgebildet.

### Eingabeseite

Name	Bezeichnung	Wert	Einheit
R_z	Zähnezahl	20	<input type="text" value=""/>
R_mn	Normalmodul	2	mm <input type="text" value=""/>
R_beta	Schrägungswinkel	11	° <input type="text" value=""/>
BP_cPstern	Kopfspielfaktor	0,25	<input type="text" value=""/>
R_x	Profilverschiebungsfaktor	0,5	<input type="text" value=""/>

Abbildung 102: Beispiel der Eingabeseite des Prozesssimulators

### Ausgabeseite

Name	Bezeichnung	Wert	Einheit
R_d	Teilkreisdurchmesser	40,749	mm <input type="text" value=""/>
R_df	Fußkreisdurchmesser	37,749	mm <input type="text" value=""/>

Abbildung 103: Beispiel der Ausgabeseite des eingebauten Prozesssimulators

Diese zwei Fenster sind Bestandteil des entwickelten Prozesssimulators, welcher die schnellen Tests des aufgebauten Ablaufs erlaubt.

Nachdem die Startwerte des Prozesses belegt wurden, kann die eigentliche Berechnung gestartet werden. Dabei werden die eingegebenen Werte in die zentrale Datenbank geschrieben. Von dort aus sind diese jedem Funktionsbaustein zugänglich. Analog dazu wird jeder berechnete Wert eines Merkmals in der Datenbank abgelegt. Somit beinhaltet die Datenbank am Ende der erfolgreichen Abarbeitung aller Prozessbeteiligten entlang des Ablaufs alle Eingangs- und Ausgangsdaten der einzelnen Funktionsbausteine. Analog entsteht die Ausgabeseite mit den Ergebnismerkmalen. Diese wird nach den abgearbeiteten Ergebnismerkmalen gefiltert und anschließend zusammen mit den entsprechenden Merkmalswerten aus der gemeinsamen Datenbank angezeigt. Die Filterung dient der

Überprüfung der Ausgabeliste, um die nicht abgearbeiteten Ergebnismerkmale (abhängig vom Ablauf) gegebenenfalls aus der Liste zu entfernen, was zur besseren Übersichtlichkeit der Ergebnisse führt. Die Abarbeitung der Funktionsbausteine entlang des Ablaufs erfolgt immer nach dem gleichen Prinzip:

1. Eingangsdaten aus dem gemeinsamen Speicher holen,
2. Funktion bzw. den Algorithmus des Funktionsbausteines mit diesen Werten ausführen,
3. Ergebnis der Berechnung in den gemeinsamen Speicher schreiben und
4. anhand der ausgehenden Kante den nächsten Prozessknoten bestimmen und Punkte 1-3 mit diesem wiederholen.

Die Abarbeitung der Gateways erfolgt analog, denn jedes Gateway hat im Grunde genommen eine eigene Funktion, die aber keine Merkmalswerte berechnet, sondern den Kontrollfluss steuert. Die Ergebnisse der Funktionen, welche in den verschiedenen Gateways hinterlegt sind, helfen anhand der entsprechenden Bedingungen zu entscheiden, welcher von den Prozessbeteiligten zum welchen Zeitpunkt abgearbeitet werden soll.

Ein Sub-Prozess ist ein Prozess, welcher in einen anderen Prozess eingebunden wird und zur Verbesserung der Übersichtlichkeit die inkludierten Elemente unter einem graphischen Symbol verbirgt.

Die Verwendung eines Teilprozesses kann mit der Anwendung einer Klassen-Methode verglichen werden. Der Aufruf eines Teilprozesses sieht genauso wie eine Methode eine endliche Anzahl der Eingabe- und Ausgabeparameter sowie den Algorithmus für die Umrechnung vor. Alles was innerhalb der Methode abläuft, bleibt dem Gesamtablauf verborgen. Außerdem können die programmierten Algorithmen bei Bedarf wiederverwendet werden.

Die Anwendung kann am Beispiel der Berechnung des Achsabstandes einer Stirnrad-Getriebestufe illustriert werden. Dieser Sub-Prozess besteht aus folgenden Funktionen:

1. Berechnen des Null-Achsabstandes  $a_d$  aus den zwei Teilkreisdurchmessern nach Gleichung (3-7)
2. Berechnen des Achsabstandes  $a$  aus dem Null-Achsabstand  $a_d$ , dem Betriebseingriffswinkel  $\alpha_{wt}$  und dem Stirneingriffswinkel nach Gleichung. (3-8)

$$a_d = f(d_1, d_2) = \frac{d_1 + d_2}{2} \quad (3-7)$$

$$a=f(a_d, a_{wt}, a_t)=a_d \cdot \frac{\cos(a_t)}{\cos(a_{wt})} \quad (3-8)$$

Die Eingabeparameter eines solchen Teilprozesses sind  $d_1$ ,  $d_2$ ,  $a_{wt}$ ,  $a_t$  und der Ausgabeparameter  $a$ . Der Nullachsabstand wird dabei wie eine interne Prozessvariable behandelt.

Bei einer mehrfachen Instanziierung (z. B. bei einem mehrstufigen Stirnradgetriebe) dieses Sub-Prozesses müssen alle Eingabeparameter jeweils für die Stufe definiert werden sowie der Achsabstand selbst. Aus diesem Grund werden auch die entsprechenden Merkmale (Ports) automatisch beim Hinzufügen des Teilprozesses auf der Arbeitsfläche des Designers erzeugt. Da aber auch die Zwischenergebnisse oft gebraucht werden, wird der Null-Achsabstand für jede Instanz eines Sub-Prozesses erzeugt und ist damit später bei der Simulation auf der Ausgabeseite zu sehen.

Bei der Generierung einer Instanz eines Funktionsbausteins erhält dieser genauso wie die entsprechenden Eingabe- und Ausgabemerkmale eine eindeutige Identifikationsnummer (ID). Diese ID ist der Schlüssel für einen fehlerfreien Zugriff auf die Prozessvariablen und die Verarbeitungsregeln (Funktionen und Algorithmen) während der Ausführung des Ablaufs. Das Abspeichern eines modellierten Ablaufs in Form einer XML-Datei inklusive aller ID's und Beziehungen, ermöglicht eine bequeme Wiederverwendung des Ablaufs, ohne diesen neu zusammensetzen zu müssen.

Etwas komplexer sieht die mehrfache Wiederverwendung eines Sub-Prozesses aus. Hierbei muss die damit verknüpfte mehrfache Instanziierung aller im Sub-Prozess beteiligten Prozesselemente (Merkmale, Funktionsbausteine und Kontrollflusselemente) eines Teilprozesses mit bedacht werden, damit keine doppelten Identifikationen bei den verschiedenen Elementen des gleichen Typs verwendet werden. Aus diesem Grund wird beim Laden eines Sub-Prozesses aus der Datenbank allen Elementen automatisch eine neue ID vergeben. Dies geschieht unter Berücksichtigung und Einhaltung aller in dem Sub-Prozess definierten Verknüpfungen zwischen den Sub-Prozess-Elementen. Somit ist in der Datenbank ein Sub-Prozess als ein sogenannter Masterprozess abgelegt.

Die Abarbeitung eines Sub-Prozesses erfolgt analog zu einem Prozess. Allerdings besitzen die von dem Sub-Prozess inkludierten Funktionsbausteine keine direkte Verbindung zu den Prozesselementen des Mainprozesses (siehe Abbildung 104).

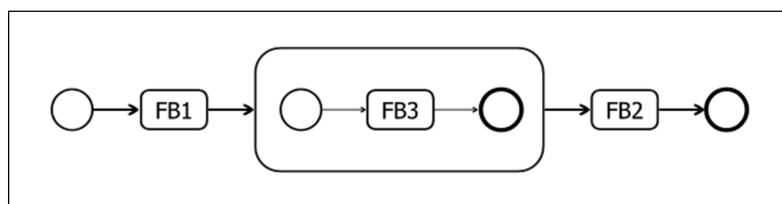


Abbildung 104: Sub-Prozess aufgeklappt

Wenn ein Sub-Prozess beim Ausführen an der Reihe ist, wird innerhalb dessen nach dem Start-Knoten gesucht. Dieser darf nur einmal vorkommen. Wurde der Start-Knoten gefunden, erfolgt die Ausführung des Prozesses innerhalb des Subprozesses nach dem gleichen Schema wie bei einem Prozess der obersten Ebene.

### 3.4.6 Generierung und Ausführung der Applikationen

Die zweitwichtigste Aufgabe des entwickelten IT-Systems liegt in der Erstellung von lauffähigen Applikationen auf Basis der modellierten Abläufe. Als erstes muss ein Berechnungsablauf mit Hilfe des Prozess-Designers modelliert werden. Die graphisch modellierten Abläufe werden in der Projektdatei im XML Format abgespeichert. Diese Datei wird an ein Werkzeug zur automatischen Erstellung von Berechnungsoberflächen weitergegeben. Wobei die XML-Dateien interpretiert und entsprechende Eingabe- und Ausgabeseite aufgebaut werden.

Die im Repository abgelegten Hilfeinformationen zu den einzelnen Merkmalen werden dabei eingefügt und können später über entsprechende Buttons bequem aufgerufen werden. Gleichzeitig wird die Projektdatei an die Prozess-Engine weitergeleitet. Diese entnimmt daraus die logische Abfolge der Prozesskette und führt dementsprechend die einzelnen Funktionsbausteine aus (siehe Kapitel 3.4.5). Nach dem Kompilieren können die Berechnungsmodule über die XML-Schnittstellen in jeder geeigneten Benutzeroberfläche zur Verfügung gestellt werden. Hier wurden als Ein- und Ausgabe-Interface die entsprechenden Werkzeuge des Berechnungssystems MDESIGN genutzt.

Die Ausführung der Applikationen beginnt mit der Eingabe der für die Berechnung notwendigen Eingangsdaten über das GUI der Applikation. Die GUI-Engine ordnet diese Eingangsdaten den entsprechenden Merkmalen in der Datenbank (gemeinsamer Speicher) zu und stellt diese somit allen Prozessbeteiligten zur Verfügung. Im Gegensatz zur klassischen Software erfolgt die Umwandlung der Eingangsdaten der Applikation in die Ergebnisse nicht mit Hilfe im Quellcode festgelegter und somit kodierter Logik, sondern auf Basis des in der Projektdatei abgelegten graphisch modellierten Berechnungsablaufes. Die Prozess-Engine interpretiert die Projektdatei und führt entsprechend die einzelnen Funktionsbausteine des Prozesses aus. Dabei werden die für einen Funktionsbaustein benötigten Eingangsdaten dem gemeinsamen Speicher entnommen und zusammen mit der Identifikationsnummer des Funktionsbausteines an die Skript-Engine geschickt. Die Skript-Engine ist für die Ausführung der in den Funktionsbausteinen hinterlegten C# Skripten zuständig. Anhand der übergebenden Identifikationsnummer des Funktionsbausteines wird von der Skript-Engine der entsprechende C#-Skript des Wissens-Repository entnommen und ausgeführt. Die entsprechende Programmarchitektur ist in der Abbildung 105 schematisch aufgeführt.



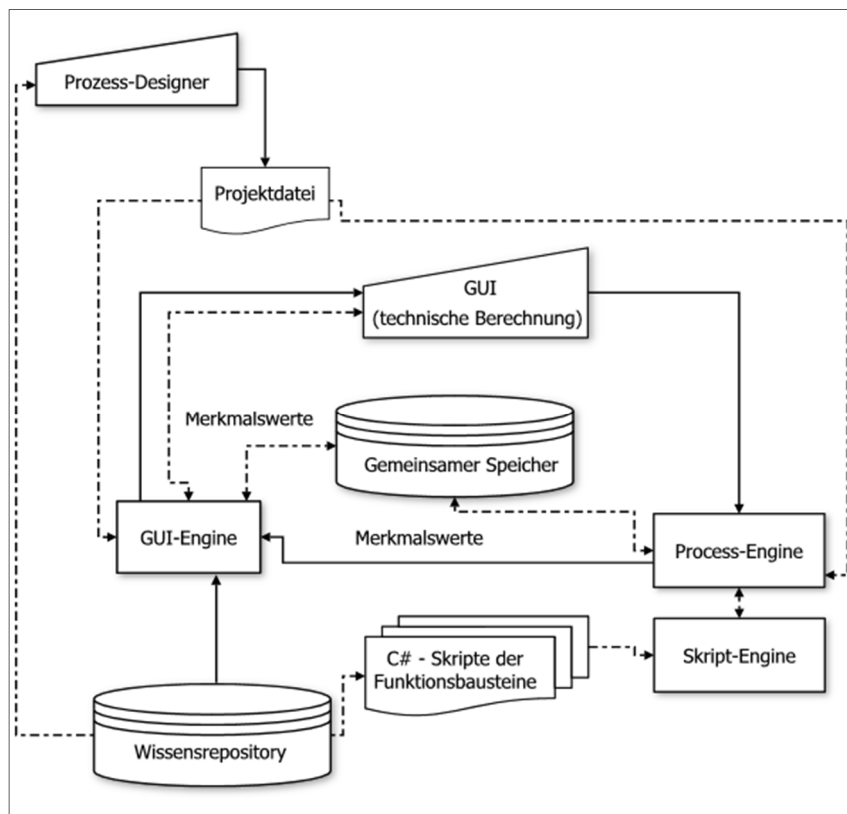


Abbildung 105: Architektur des entwickelten PM Systems

Nachdem die Skript-Engine die Ergebnisse der Berechnung an die Prozess-Engine übergeben hat, werden diese auch in dem gemeinsamen Speicher abgelegt. Ist das Ende des Prozesses erreicht, stößt die Prozess-Engine die GUI-Engine an, welche die errechneten Merkmalswerte zur Interpretation auf der Ausgabeseite der Applikation darstellt. Ein Beispiel der ausgeführten Applikation für die Verzahnungsberechnung ist in der nachfolgenden Abbildung dargestellt.

### 3.4.7 Fazit

Auf Basis der im ersten Teil theoretisch erarbeiteten Prinzipien wurde ein Beispiel für die praktische Umsetzung des Konzepts „Prozessgebundene Berechnungsbaugruppen“ dargestellt. Die Umsetzung basiert auf den formalen visuellen Modellen der Organisation eines Berechnungsprozesses. Diese wurden in dem entwickelten Design Process Management System implementiert. Ein speziell dazu entwickelter graphischer Editor (Prozess-Designer) ermöglicht dabei die Modellierung der Berechnungsprozesse. Das entwickelte DPM System zeichnet sich durch eine systematische und übersichtliche Modellierung der Berechnungsabfolgen aus. Es können sequentielle sowie parallele Prozesse modelliert werden, welche abhängig von der Aufgabenstellung Verzweigungen und Zusammenführungen aufweisen können. Die Ablage der modellierten Abläufe in Form von Teilprozessen in dem Wissens-Repository sichert die Wiederverwendung der standardisierten sowie der „Best Practice“ Lösungen. Durch die graphische Notation werden nicht nur die Ergebnisse eines Berechnungsablaufes, sondern auch deren Entstehungsablauf dokumentiert. Die mögliche

Teilautomatisierung des Berechnungsprozesses sorgt dabei für die Verkürzung der Entwicklungszeiten sowie Einheitlichkeit bei der Lösungsdefinition. Zu der weiteren Besonderheit des entwickelten Systems gehört die Generierung von ausführbaren Applikationen, basierend auf den modellierten Workflows. Im nächsten Kapitel soll die Anwendung des entwickelten Ansatzes an einem Berechnungsprozess dargestellt werden.

## 4 Anwendung des Ansatzes am Beispiel Getriebeberechnung

### 4.1 Theoretische Grundlagen des Berechnungsbeispiels

In diesem Kapitel wird der ausgearbeitete Ansatz „Prozessgebundene Berechnungsbaugruppen“ einer praktischen Prüfung am Beispiel Getriebeberechnung unterworfen. Als Aufgabe wird die Auslegung eines Industrie-Stirnradgetriebes gewählt, welches folgenden Anforderungen genügen soll:

Antrieb:	E-Motor
Antriebsleistung P:	20 kW
Antriebsdrehzahl n:	1500 1/min
Gesamtübersetzungsverhältnis i:	12
Maximale Abweichung der Übersetzung $\Delta i$ :	$\pm 2,5 \%$
Anwendungsfaktor $K_A$	1,25
Schrägungswinkel $\beta$	8-20 °
Verzahnungsqualität q	6

Tabelle 13: Daten des zu entwickelnden Beispielgetriebes

Der Getriebeentwicklungsprozess untergliedert sich in die Entwicklung einzelner Getriebeelemente wie Verzahnung, Lagerung, Wellen etc. Die Verzahnungen gehören zu den anspruchsvollsten Komponenten eines Getriebes, diese können in verschiedenen Formen vorkommen (Stirnräder, Kegelräder, Schneckenräder, etc.) sowie diverse Anordnungen (Planetengetriebe, Stirnradgetriebe) aufweisen. Demzufolge wird das Hauptaugenmerk in diesem Beispiel auf die Auslegung der Verzahnung gelegt.

#### 4.1.1 Verzahnungsberechnung

Für die Auslegung der Verzahnung existiert mittlerweile eine Reihe von Berechnungsvorgehensweisen bzw. Vorschlägen. Im Rahmen dieser Arbeit wird nicht auf die fachlichen Aspekte der angewendeten Berechnungsverfahren eingegangen, sondern auf ihre Implementierung mit dem im Rahmen dieser Arbeit aufgestellten Ansatz unter Einschluss des entwickelten DPM Systems. Bei den nachfolgenden Berechnungen wird Bezug genommen auf den fachlichen Hintergrund und Sachverhalt bei (Schlecht, 2009, Römhild, 1993, Linke, 2012, Niemann und Winter, 2005, Dyla, 2002, Jaroš, 2007, Bulligk, 2009, DIN 3990-3:1987-12, DIN 3960:1987-03, ISO 6336-3:2006-09).

Bevor die eigentliche Auslegung eines Getriebes beginnt, müssen die Randbedingungen geklärt werden: Getriebeart, Anzahl und Anordnung der Stufen (Jaroš, 2007). Die Art des Getriebes richtet sich nach der gewünschten Achsanordnung, der zu übertragenden Leistung, Antriebsdrehzahl etc. Im aufgeführten Beispiel ist die Art des Getriebes als Anforderung vorgegeben. Für die Anzahl der Stufen existieren im Wesentlichen zwei Empfehlungen: (Niemann und Winter, 1983) und (Römhild, 1993). Da

diese sich nur wenig unterscheiden wird der modernere Ansatz für eine massenminimierte Getriebeauslegung nach (Römhild, 1993) genommen:

Gesamtübersetzung $\leq 5$	$5 < \text{Gesamtübersetzung} \leq 15$	$15 < \text{Gesamtübersetzung}$
1 Stufe	2 Stufen	3 Stufen

Tabelle 14 : Abhängigkeit der Anzahl der Stufen von der Gesamtübersetzung nach (Römhild, 1993)

Diese Schätzregeln können als Umrechnung der Gesamtübersetzung  $i_{\text{ges}}$  in die Anzahl der Stufen  $n$  interpretiert werden. In der Klassifikation nach eCl@ss entsprechen diese Größen den Merkmalen der Klasse Stirnradgetriebe (23-32-01-01). Somit kann ein neuer Funktionsbaustein „ $n(i_{\text{ges}})$ “ nach Römhild“ angelegt und dem Merkmal „ $n$  - Anzahl der Stufen“ zugeordnet werden. Im weiteren Verlauf dieses Beispiels müssen analog für alle Funktionen und Formeln entsprechende Merkmale und Funktionsbausteine definiert und dann, falls in dem Wissens-Repository nicht vorhanden, neu angelegt werden. Bei den nachfolgenden Berechnungsroutinen werden diese Schritte nicht mehr explizit beschrieben. Die Liste der für dieses Beispiel angelegten Merkmale und der entsprechenden Funktionsbausteine kann dem Anhang A dieser Arbeit entnommen werden.

Als nächstes muss die Aufteilung der Gesamtübersetzung durchgeführt werden. Für eine massenoptimierte Getriebeauslegung nach (Römhild, 1993) wird diese gemäß der Tabelle 15 bestimmt.

1 Stufe	2 Stufen	3 Stufen
$u_1 \approx u$	$u_1 \approx 0,7332 \cdot u^{0,6438}$	$u_1 \approx 0,4643 \cdot u^{0,609}$
	$u_2 = \frac{u}{u_1}$	$u_2 \approx 1,205 \cdot u^{0,262}$
		$u_3 = \frac{u}{u_1 \cdot u_2}$

Tabelle 15: Aufteilung der Gesamtübersetzung nach (Römhild, 1993)

Als generelles Auslegungsverfahren wird der Ansatz von (Jaroš, 2007) genommen, welcher bereits den Eingang in die Praxis (nachfolgende Arbeiten und FVA Software) gefunden hat und für die rechnerbasierte Umsetzung im Vergleich zu dem Vorgehen nach (Niemann und Winter, 2005) oder nach (Linke, 2012) am besten geeignet ist.

Die Auslegungsroutinen möglichst an den Nachrechnungsroutinen zu orientieren (Jaroš, 2007) beziehungsweise aus ihnen abzuleiten, verbessert die Durchgängigkeit des Berechnungsprozesses.

Ausgangspunkt der Auslegung von Getriebedaten sind die Angaben zu den auftretenden Belastungen, gewünschten Sicherheitsfaktoren sowie diversen Kennwerten aus den Tragfähigkeitsnachweisen. Anschließend können durch Umstellen entsprechender Gleichungen für die Nachrechnung die Grunddaten der Verzahnungsgeometrie hergeleitet werden (siehe Abbildung 106).

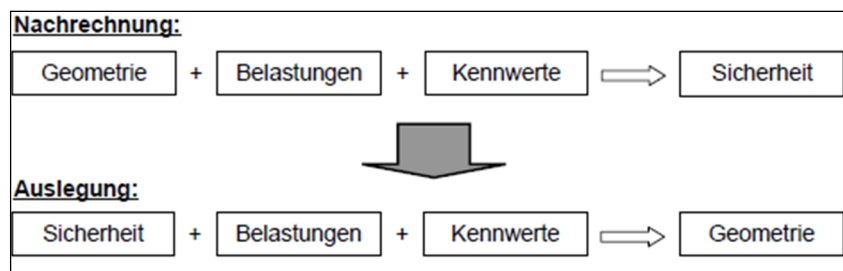


Abbildung 106: Ableitung der Auslegungsroutinen aus den Nachrechnungsalgorithmen (Jaroš, 2007)

Da die Kennwerte zu Beginn der Auslegung noch nicht vorliegen, werden diese mit Startwerten belegt. Die in dem Auslegungsschritt ermittelte Verzahnungsgeometrie wird anschließend der Nachrechnung unterzogen, bei der die mit Startwerten belegten Faktoren und Kennwerte konkretisiert und an die Geometrie angepasst werden. Die abwechselnde Durchführung der Auslegung und der Nachrechnung führt zu einem iterativen Prozess, an dessen Ende die Verzahnungsgeometrie gezielt auf die gewünschte Sicherheit abgestimmt ist. Der schematische Ablauf des Auslegungsprozesses ist in der Abbildung 107 dargestellt.

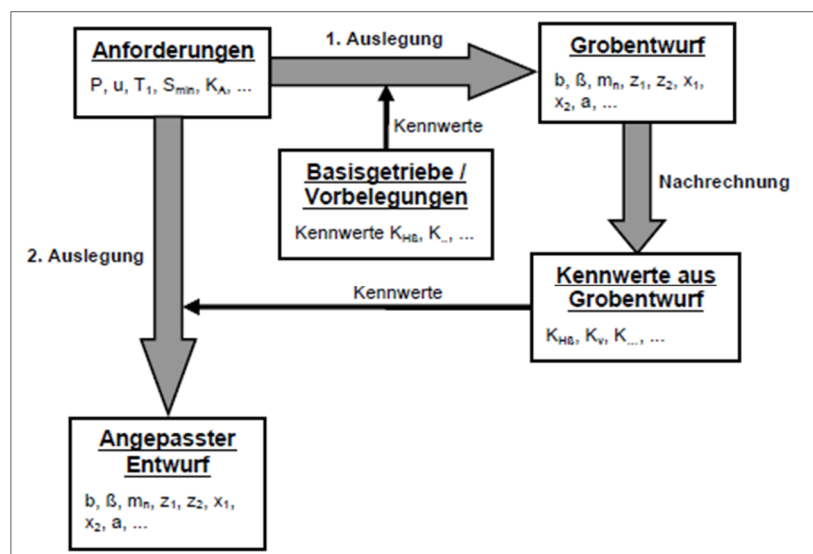


Abbildung 107: Iterative Vorgehensweise der Auslegung nach (Jaroš, 2007)

In der Abbildung 109 ist der generelle Ablauf für die Auslegung der wesentlichen Geometrieparameter eines Zahnradpaares nach (Jaroš, 2007) dargestellt. Hier findet gleichzeitig die Auslegung auf die Sicherheit gegen Zahnfußbruch und gegen Grübchenbildung statt. Dabei sollen die beiden Zielsicherheiten ( $S_{Fmin}$  und  $S_{Hmin}$ ) den Wert von ca. 1.3 aufweisen. Dies erfordert aber eine gezielte Optimierungsmaßnahme, denn typischerweise ist bei einem Verzahnungsgeometriesatz die Sicherheit gegen Zahnfußbruch höher als die gegen Grübchenbildung.

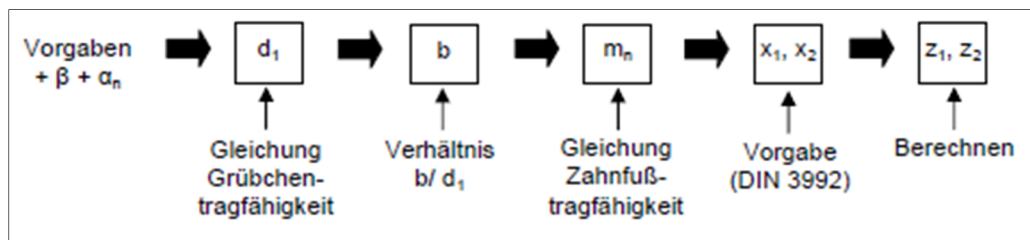


Abbildung 108: Ablauf der Erstausslegung der Verzahnung (ebd.)

Aus diesem Grund wird hier die Auslegung in Bezug auf die typischerweise kleinere Sicherheit gegen die Grübchenbildung durchgeführt. Der schematische Ablauf ist aus Abbildung 109 zu erkennen.

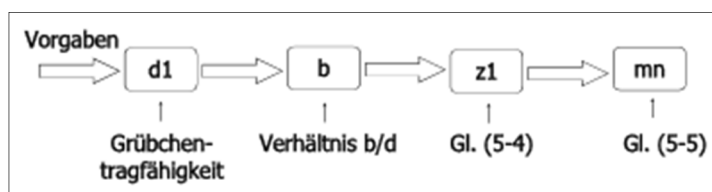


Abbildung 109: Ablauf der Auslegung

Als erstes soll der Teilkreisdurchmesser ermittelt werden, danach aus dem gewünschten Verhältnis des Teilkreisdurchmessers zur Breite die Breite des Zahnrades und anschließend die Zähnezah und der Normalmodul. Für die Bestimmung des Teilkreisdurchmessers wird die Gleichung für die Berechnung der Sicherheit gegen Grübchenbildung nach DIN 3990-2:1987-12 bzw. ISO 6336-2:2006-09 unter Einführung des Verhältnisses Breite zum Durchmesser (siehe Gleichung (4-1)) folgendermaßen umgestellt:

$$b = \left( \frac{b}{d_1} \right) \cdot d_1 \quad (4-1)$$

$$d_1 \geq \sqrt[3]{\left( \frac{b}{d_1} \right) \cdot \frac{u+1}{u} \cdot \left( \frac{S_{H \min}}{\sigma_{H \lim}} \cdot \frac{Z_B \cdot Z_H \cdot Z_E \cdot Z_\epsilon \cdot Z_\beta \cdot \sqrt{K_A \cdot K_V \cdot K_{H\beta} \cdot K_{H\alpha}}}{Z_{NT} \cdot Z_L \cdot Z_V \cdot Z_R \cdot Z_W \cdot Z_X} \right)^2} \quad (4-2)$$

Bei der ersten Auslegung sind die Z- und K-Faktoren noch unbekannt, aus diesem Grund werden diese am Anfang mit den Startwerten belegt. Die Startwerte für die Faktoren in dieser Gleichung sind in der nachfolgenden Tabelle nach (Jaroš, 2007) aufgelistet.

Parameter	Benennung	Startwert
$b/d_1$	Breiten-/Durchmesser Verhältnis	0,6 ... 1,4
$SH_{min}$	Mindestsicherheit gegen Grübchen	1,3
$Z_B$	Ritzel-Eingriffsfaktor	1,0
$Z_H$	Zonenfaktor	2,5
$Z_E$	Elastizitätsfaktor	$Z_E = \sqrt{\frac{1}{\pi \cdot \left( \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} \right)}}$
$Z_\epsilon$	Überdeckungsfaktor	1,0
$Z_\beta$	Schrägenfaktor	$Z_\beta = \sqrt{\cos(\beta)}$
$K_A$	Anwendungsfaktor	1
$K_V$	Dynamikfaktor	1,1
$K_{Ha}$	Stirnfaktor	1,2
$K_{H\beta}$	Breitenfaktor	1,5
$Z_{NT}$	Lebensdauerfaktor	1,0
$Z_L, Z_v, Z_R$	Einflussfaktoren für die Schmierfilmbildung	1,0
$Z_W$	Werkstoffpaarungsfaktor	1,0
$Z_X$	Größenfaktor	1,0

Tabelle 16: Startwerte für die Faktoren in Gleichung (4-2), (vgl. ebd.)

Diese Startwerte werden im weiteren Verlauf des Berechnungsprozesses iterativ präzisiert. Die Bestimmung der Zähnezahl des Ritzels  $z_1$  erfolgt nach dem Vorschlag von (Niemann und Winter, 2005) für die ausgeglichene Fuß- und Grübchentragfähigkeit (siehe Gleichung (4-3)).

$$z_1 = 35000 \cdot \frac{u+1}{u} \cdot \frac{\sigma_{FE}/S_{Fmin}}{\sigma_{Hlim}/S_{Hmin}} \cdot \cos(\beta)^2 \cdot \frac{f_L}{Z_W^2} \quad (4-3)$$

Wobei  $Z_W = 1$  (Werkstofffaktor für Ritzel und Rad mit gleichen Fuß- und Grübchentragfähigkeiten),  $f_L = 1$  steht für schwellende, bzw.  $f_L = 1.1$  für wechselnde Belastung (zum Beispiel bei Zwischenrädern) (Niemann und Winter, 2005). Mit der vorhandenen Zähnezahl und dem Teilkreisdurchmesser kann anschließend der Normalmodul  $m_n$  nach Gleichung (4-4) wie folgt ermittelt werden:

$$m_n = \frac{d_1 \cdot \cos(\beta)}{z_1} \quad (4-4)$$

Die Profilverschiebungsfaktoren können generell entweder eingegeben oder bei einer Geradverzahnung mit Hilfe des Vorschlages nach (Römhild, 1993) ermittelt werden. In Ihrer Dissertation hat Römhild für die optimale Summe der Profilverschiebungsfaktoren (in Hinsicht auf die minimale Masse des Getriebes) folgende Gleichungen hergeleitet:

$$(x_1 + x_2)_{\text{opt}} = 0,3117 \cdot u^{0,3952} \cdot z_1^{(0,0046 \cdot u^2 - 0,037 \cdot u + 0,5575)} \quad (4-5)$$

$$(x_1 + x_2)_{\text{opt}} = (0,0718 \cdot u + 0,2448) \cdot z_1^{(-0,0123 \cdot u + 0,538)} \quad (4-6)$$

Diese Gleichungen sind nur für die Geradverzahnung gültig, wobei mit Gleichung (4-5) für die Übersetzung ins Schnelle ( $i < 1$ ) und mit Gleichung (4-6) für die Übersetzung ins Langsame die Summe der Profilverschiebungsfaktoren  $\Sigma x$  ermittelt werden kann.

Aus den ermittelten Größen wie Zähnezahzahl des Ritzels  $z_1$  und Übersetzung der ersten Stufe kann die Zähnezahzahl des Rades der ersten Stufe sowie dessen Teilkreisdurchmesser  $d_1$  errechnet werden. Mit den vorhandenen Teilkreisdurchmessern des Ritzels  $d_1$  und des Rades  $d_2$  ergibt sich der Achsabstand nach Gleichung (4-7)

$$a_d = \frac{d_1 + d_2}{2} \quad (4-7)$$

Die ermittelten Grunddaten der Verzahnung ermöglichen einen anschließenden Nachweis gegen die Grübchenbildung, welcher die anfangs mit Startwerten belegten geometrieabhängigen Faktoren präzisieren lässt. Die wiederholte Auslegung mit den neu errechneten Kennwerten liefert mit großer Wahrscheinlichkeit einen abweichenden Wert des Teilkreisdurchmessers  $d_1$  nach Gleichung (4-2). Somit sind oft mehrere Iterationen notwendig, um den der geforderten Sollsicherheit gegen die Grübchenbildung  $S_{H\min}$  passenden Geometriesatz zu bestimmen.



## 4.2 Umsetzung mit dem Prozess-Designer

Die Umsetzung des im obigen Kapitel beschriebenen Auslegungsverfahrens mit dem im Rahmen dieser Arbeit entwickelten Workflow Management System gliedert sich in mehrere Phasen. Zuerst müssen alle in den Berechnungen beteiligten Parameter in Form von Produktmerkmalen in Anlehnung nach eCl@ss identifiziert werden. Wenn diese Merkmale noch nicht im Merkmals-Repository vorhanden sind, müssen sie angelegt werden. Auf der Abbildung 110 ist auf der linken Seite der Auszug aus der Liste der Funktionsbausteine für die Klasse Stirnradsatz zu sehen, welche im Rahmen dieses Beispiels angelegt wurde.

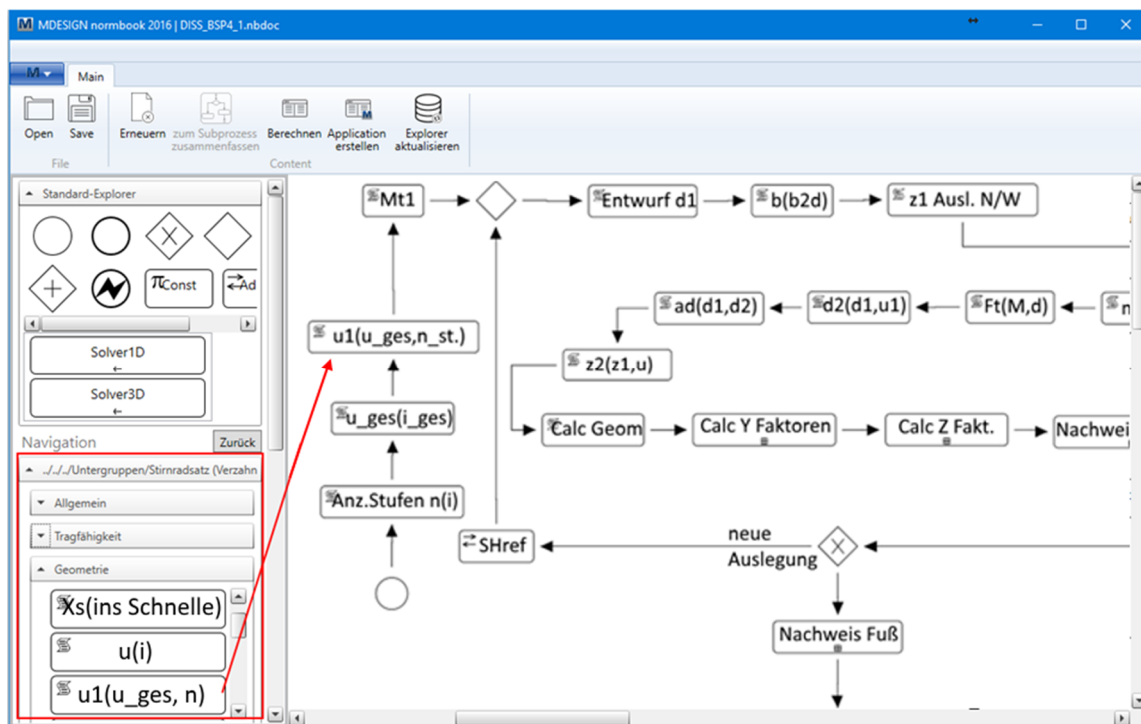


Abbildung 110: Modellieren mit dem Prozess-Designer

Die Merkmale und entsprechend die Funktionsbausteine sind in Gruppen wie „Geometrie“, „Tragfähigkeitsnachweis-Fuß“ oder „Tragfähigkeitsnachweis-Flanke“ strukturiert. Mittels Drag & Drop Technik können die Funktionsbausteine aus dem Wissens-Repository auf die Arbeitsfläche gezogen und entsprechend platziert werden. Nach der Verbindung der Funktionsbausteine auf der Arbeitsfläche können bei Bedarf die Prozessvariablen kontrolliert und gegebenenfalls editiert werden. In der Abbildung 111 sind beispielsweise die Eigenschaften des Funktionsbausteines für die Ermittlung der Teilübersetzung der ersten Stufe nach Tabelle 15 dargestellt. Merkmale sowie ausgewählte Funktionsbausteine, die für dieses Beispiel notwendig sind, finden sich nochmals im Anhang A.



Abbildung 111: Angabe der Prozessvariablen mit dem Prozess-Editor

Bei der Untergliederung des Auslegungsprozesses in die Teilaufgaben sind einzelne Teilprozesse entstanden. Diese sind für die eigentliche Auslegung der Geometrie der ersten Stufe (für das Ritzel und das Rad) sowie die Bestimmung der Z-Faktoren und den anschließenden Tragfähigkeitsnachweis gegen die Grübchenbildung zuständig. Der modellierte Prozess der Auslegung der ersten Stufe ist in der Abbildung 112 dargestellt.

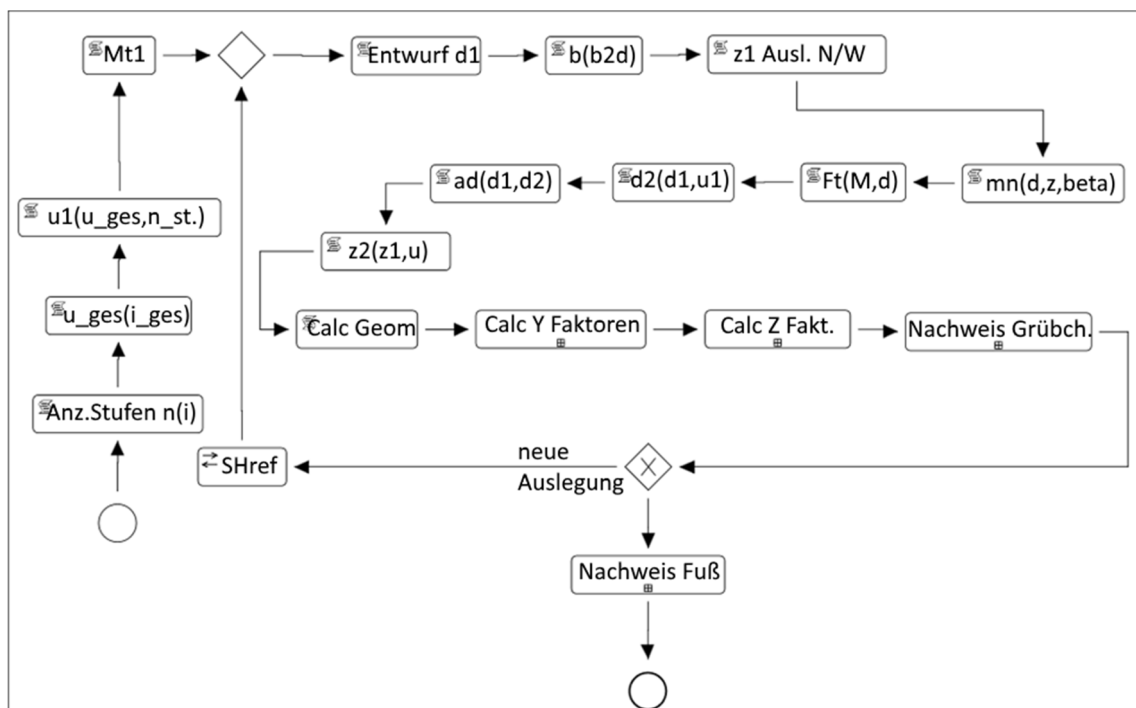


Abbildung 112: Ablauf der modellierten Auslegung von Stufe 1

Die am Anfang ermittelte Anzahl der Stufen und die davon abhängige Aufteilung der Gesamtübersetzung fließt in die Berechnung des Antriebsmomentes  $M_{t1}$ . Danach folgt die Definition

der Konstanten, welche die Startwerte der Faktoren für den Nachweis der Grübchentragfähigkeit beinhaltet. Die anschließende iterative Anpassung der Verzahnungsgeometrie wiederholt sich solange, wie die Abbruchbedingungen noch nicht in Kraft treten. Die erste Abbruchbedingung beschreibt dabei die Unterschreitung der Differenz  $\Delta S_H$  zwischen dem Wert der geforderten Sollsicherheit  $S_{Hmin}$  für die Bestimmung des Teilkreisdurchmessers des Ritzels  $d_1$  und dem Wert, der beim Nachweis der Grübchentragfähigkeit und der dabei bestimmter tatsächlicher Sicherheit  $S_{Htats}$  (siehe Gleichung (4-8)) ermittelt wird.

$$\Delta S_H = |S_{Hmin} - S_{Htats}| \quad (4-8)$$

Die zweite Abbruchbedingung beschreibt die Anzahl der möglichen Iterationen beziehungsweise die Veränderung der im Tragfähigkeitsnachweis bestimmenden Sicherheit  $S_{Htats}$ . Soll der Wert der Sicherheit  $S_{Htats}$  sich nicht mehr von dem Wert aus dem vorangegangenen Berechnungsschritt  $S_{Href}$  unterscheiden oder ist die Anzahl der Iterationsschritte höher als die maximal erlaubte Anzahl, wird die Auslegung unterbrochen. Der Ablauf der Berechnung der tatsächlichen Sicherheit gegen Grübchenbildung ist als Sub-Prozess im Wissens-Repository abgelegt. In der Abbildung 113 ist dieser aufgeklappt dargestellt.

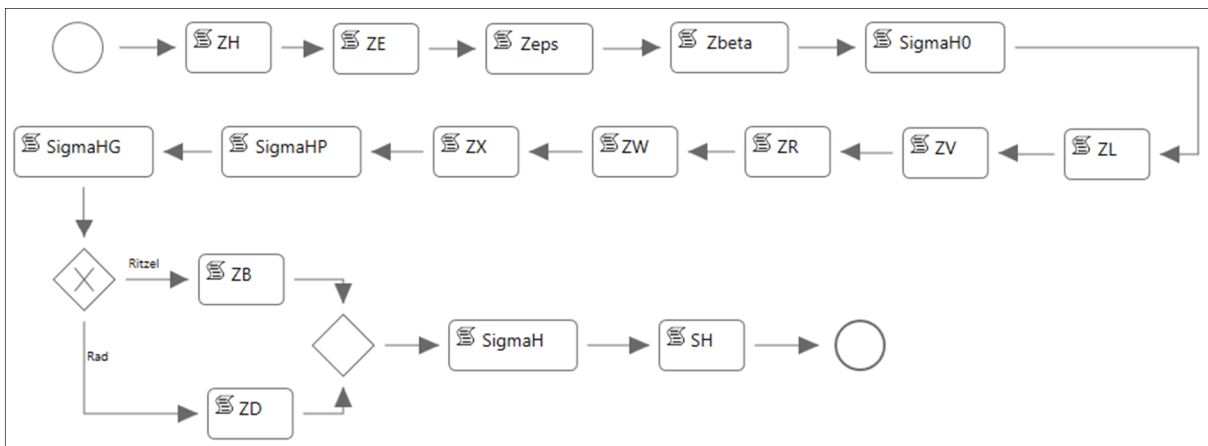


Abbildung 113: Ablauf der modellierten Berechnung der Flankensicherheit SH

Mit dem in der Abbildung 112 dargestellten Berechnungsablauf werden die Werte des Verzahnungssatzes der ersten Stufe ermittelt. In der Tabelle 17 sind die Ergebnisse der Auslegung der ersten Stufe inklusive der beiden Tragfähigkeitsnachweise (für den Zahnfuß und die Zahnflanke) aufgelistet.

<b>Antrieb</b>				
Drehmoment	$M_{T1}$	=	127,324	N*m
Antriebsdrehzahl	$n_1$	=	1500	1/Umdr.
<b>Verzahnung</b>				
Teilkreisdurchmesser	$d_1$	=	48,917	mm
Zahnbreite	$b$	=	39,13	mm
Zähnezahl	$z_1$	=	24	-
Normalmodul	$m_n$	=	2,038	mm
Teilkreisdurchmesser2	$d_2$	=	177,325	mm
Zähnezahl2	$z_2$	=	87	-
Flankensicherheit	$S_{H1}$	=	1,316	
Vorhandene Flankenpressung	$\sigma_{H1}$	=	887,937	N/mm <sup>2</sup>
Vorhandene Zahnfußspannung	$\sigma_{F1}$	=	190,43	N/mm <sup>2</sup>
Zahnfußsicherheit	$S_{F1}$	=	4,357	
Minimale Flankensicherheit	$S_{Hmin}$	=	1,3	
<b>Stirnradsatz (Verzahnungselement)</b>				
Umfangskraft am Teilkreis	$F_{t1}$	=	5205,698	N
Null-Achsabstand	$a_0$	=	113,262	mm
Übersetzung	$i_1$	=	3,625	

Tabelle 17: Ergebnisse der Auslegung der ersten Stufe

Das gesetzte Ziel, nämlich einen Verzahnungssatz zu ermitteln, dessen Sicherheit gegen Grübchenbildung  $S_H$  sich nur minimal von dem geforderten Wert  $S_{Hmin} = 1,3$  unterscheidet, wurde erreicht. Die Anpassung des Normalmoduls  $m_n$  an die Normreihe nach DIN 780 wurde nicht vorgenommen, da sonst zusätzliche Optimierungsmaßnahmen zu treffen wären, um die Sicherheit nicht unter den minimalen Grenzwert fallen zu lassen.

Die Bestimmung der Daten der zweiten Stufe erfolgt im Prinzip analog zur ersten Stufe, außer dass die Antriebsdaten entsprechend der Übersetzung der ersten Stufe umgerechnet werden müssen. Um die Wiederverwendung der erfolgreich verwendeten Bausteine und Prozesse zu demonstrieren, wurde der Teil des Ablaufes der Auslegung der ersten Stufe als Sub-Prozess im Wissens-Repository abgelegt. Dieser Teilprozess ist in der Abbildung 114 rot gekennzeichnet.

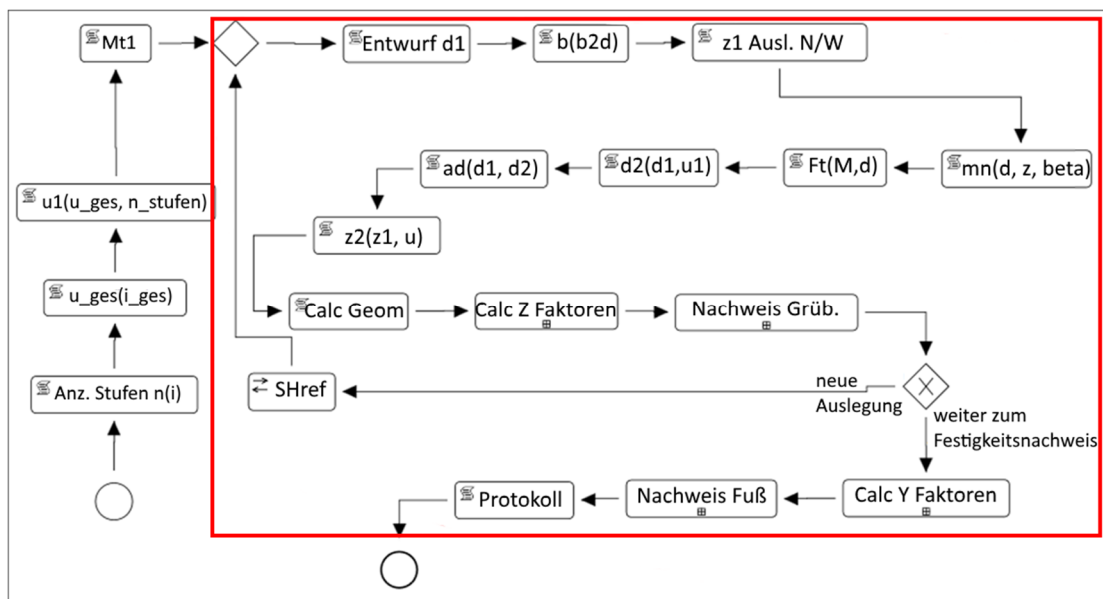


Abbildung 114: Teilprozess für die Auslegung einer Verzahnungsstufe

Somit lässt sich die Auslegung der zweiten Stufe mit minimalem Aufwand realisieren. Dazu müssen lediglich die Berechnungen der Übersetzung und entsprechender Lastdaten hinzumodelliert werden. Die Abbildung 115 stellt den Auslegungsablauf für ein Stirnradgetriebe mit bis zu zwei Stufen dar.

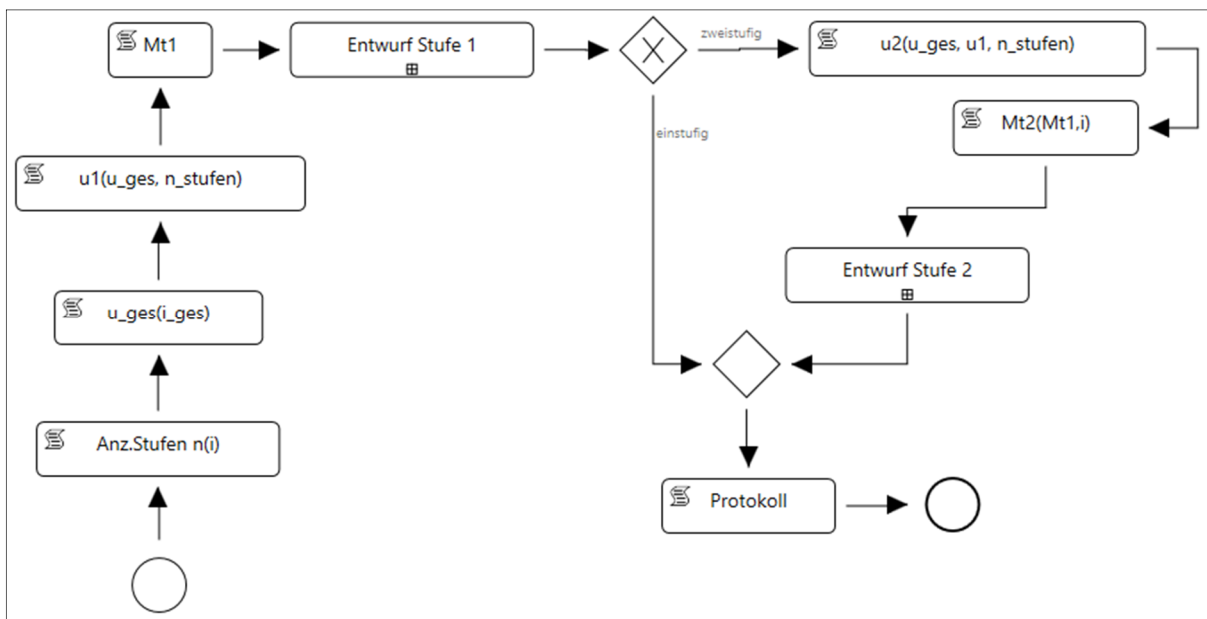


Abbildung 115: Ablauf der Auslegung der beiden Getriebestufen

Die Ergebnisse der Auslegung der zweiten Getriebestufe sind in Tabelle 18 aufgelistet.

<b>Antrieb</b>				
Drehmoment	$M_{T1}$	=	420,816	N*m
Antriebsdrehzahl	$n_1$	=	453,847	1/Umdr.
<b>Verzahnung</b>				
Teilkreisdurchmesser	$d_1$	=	73,372	mm
Zahnbreite	$b$	=	58,698	mm
Zähnezahl	$z_1$	=	24	-
Normalmodul	$m_n$	=	3,057	mm
Teilkreisdurchmesser2	$d_2$	=	241,517	mm
Zähnezahl2	$z_2$	=	79	-
Flankensicherheit	$S_{H1}$	=	1,345	
Vorhandene Flankenpressung	$\sigma_{H1}$	=	865,07	N/mm <sup>2</sup>
Vorhandene Zahnfußspannung	$\sigma_{F1}$	=	184,341	N/mm <sup>2</sup>
Zahnfußsicherheit	$S_{F1}$	=	4,512	
Minimale Flankensicherheit	$S_{Hmin}$	=	1,3	
<b>Stirnradsatz (Verzahnungselement)</b>				
Umfangskraft am Teilkreis	$F_{t1}$	=	11470,684	N
Null-Achsabstand	$a_0$	=	157,940	mm
Übersetzung	$i_1$	=	3,292	

Tabelle 18: Ergebnisse der Auslegung der zweiten Stufe

Die für die zweite Stufe ausgelegte Verzahnungsgeometrie erfüllt ebenfalls die Forderung nach der Einhaltung der minimalen Flankensicherheit  $S_{Hmin}$  von 1,3 und ist mit dem Wert von 1,345 nur unwesentlich höher. Die maximale Abweichung der Gesamtübersetzung  $\Delta i_{soll}$  von  $\pm 2,5$  % wurde mit  $\Delta i_{tats} = 0,55$  % ebenso eingehalten. Die anschließende Auswahl der Normalmodule  $m_{n1}$  und  $m_{n2}$  aus der Normreihe nach (DIN 780-1:1977-05 führt in diesem Beispiel zu den minimalen Einflüssen auf die Geometrie und kann noch über die Profilverschiebung ausgeglichen werden. Bei höheren Differenzen des ermittelten Moduls sowie der Einhaltung der Normreihe sind weitere Optimierungen notwendig, welche zum Beispiel über die Anpassung der Zahnbreite oder des Schrägungswinkels realisiert werden können.

#### 4.2.1 Beurteilung der Ergebnisse

Auf Basis der theoretischen Grundlagen aus dem Kapitel 4.1.1 wurde ein Beispiel für einen Auslegungsprozess von Stirnradgetrieben mit bis zu zwei Stufen modelliert. Die bereits für die Auslegung der ersten Stufe aufgebaute Wissensbasis wurde in Form von Merkmalen und entsprechenden Funktionsbausteinen sowie Teilprozessen im Wissens-Repository abgelegt, so dass nicht nur die elementaren Berechnungsgleichungen, sondern auch komplette Teillösungen erfolgreich Wiederverwendung fanden. Die graphische Dokumentation der Berechnungswege liefert dabei die notwendige Transparenz der Abläufe, so dass diese leicht zu erfassen sind. Der modulare Aufbau des Wissens-Repository ermöglicht eine schnelle Modifikation und Optimierung der bestehenden Abläufe und sorgt dabei für die Erhöhung der notwendigen Dynamik bei der Handhabung der Prozesse. Die automatisch erstellte graphische Oberfläche in Form eines vollständigen Berechnungsmodul (siehe

Abbildung 116) ergänzt die modellierten Abläufe mit einer interaktiven Menüführung sowie einer einheitlichen Eingabe und Ausgabe der Berechnungsinformationen. Die für die Merkmale im Wissens-Repository abgelegten Hilfeinformationen auf der Eingabeseite des Berechnungsprogramms sind nach zeitgemäßen Usability Standards konzipiert und darauf gerichtet ein schnelles Verständnis der einzugebenden Werte und Randbedingungen zu bewirken.

The screenshot displays the 'Test Objekt DPM' application window. The interface is divided into several sections:

- Top Bar:** Includes a menu bar (Berechnung, Werkzeuge, Ansicht, Analyse, Benutzerdefinierte Buttons) and a language/style dropdown (Deutsch (German), Farbstil).
- Toolbar:** Contains icons for Berechnen (F10), Zwischenablage, Berechnung schließen, Letztes Berechnungsmodul, Projektdaten, Daten importieren, Dokument, and CAD Export/Import.
- Input Section (Eingabeseite):**
  - Stirradgetriebe:** Gesamtübersetzung = 12.
  - Elektrischer Antrieb:** Leistung  $P = 20000$  W, Drehzahl  $n = 1500$  1/min.
  - Stirrad (Verzahnungselement):**
    - Verhältnis  $b/d = 0,3$
    - Lebensdauerfaktor  $Z_{NT} = 1$
    - Schmierstofffaktor  $Z_L = 1$
    - Geschwindigkeitsfaktor  $Z_V = 1$
    - Rauheitsfaktor  $Z_R = 1$
    - Werkstoffpaarungsfaktor  $Z_W = 1$
    - Größenfaktor  $Z_X = 1$
    - Flankensicherheit min.  $S_{Hmin} = 1,33$
    - Eingriffsfaktor  $Z_B = 1$
    - Zahnfußsicherheit min.  $S_{Fmin} = 1,3$
    - Schrägungswinkel  $\beta = 0^\circ$
- Output Section (Ausgabeseite):**
  - Stirradgetriebe:** Anzahl Stufen = 2, Zahnzahlverhältnis = 12.
  - Stirradsatz (Verzahnungselement):**
    - Zahnzahlverhältnis Stufe 1  $u_1 = 3,63077505$
    - Umfangskraft am Teilkreis Stufe 1  $F_{t1} = 0,03281221$
    - Null-Achsabstand Stufe 1  $a_{d1} = 299,4863142$
    - Schrägenfaktor  $Y_\beta = 1$
    - Breitenfaktor (Zahnfuß)  $K_{F\beta} = 1,129$
    - Stirnfaktor (Zahnfuß)  $K_{Fa} = 1$
    - Zonenfaktor  $Z_H = 2,462$
    - Elastizitätsfaktor  $Z_E = 193,914$
    - Schrägenfaktor  $Z_\beta = 1$
    - Breitenfaktor (Zahnflanke)  $K_{H\beta} = 1,147$
    - Stirnfaktor (Zahnflanke)  $K_{Ha} = 1$
- Help Section (Schrägungswinkel):**

**Schrägungswinkel**

Der Schrägungswinkel  $\beta$  ist durch die Tangente der Flankenlinie auf dem Teilzylinder der Radachse gegeben. Üblich sind  $\beta = 10...30^\circ$ ; bei  $\beta < 10^\circ$  sind die Vorteile der Schrägverzahnung (z.B. bessere Laufruhe) nur gering, bei  $\beta > 30^\circ$  werden die Axialkräfte ungünstig groß.

Für den Festigkeitsnachweis ist nach DIN 3990 der Schrägungswinkel  $\beta$  auf  $\beta \leq 30^\circ$  eingeschränkt, da nur hierfür die gleichen Kraftfaktoren wie bei Geradstirnrädern gelten.

Geltungsbereich für Schrägungswinkel  $\beta$ :  $0^\circ...60^\circ$ .
- Diagram Section:** Shows three cross-sectional diagrams of gears: 'Gerad' (straight), 'Schräg' (helical), and 'Doppelschräg / Stirnräder' (double helical/spur). The helical gear diagrams show the helix angle  $\beta$  and the face width  $B$ . The spur gear diagram shows the face width  $B$ .

Abbildung 116: Automatisch erstellte Applikation für die Auslegung eines Getriebes

## 5 Zusammenfassung und Ausblick

### 5.1 Zusammenfassung

Interdisziplinäre Produktanforderungen, eine hohe Servicequalität und ein verändertes Nutzerverhalten führen in immer mehr Unternehmen von klassischen funktionsorientierten und abteilungsgebundenen Organisationsformen zu einer Prozessorientierung, in der das Abteilungsdenken durch das Erfolgsziel des Gesamtprozesses ersetzt wird. Dies gilt insbesondere für einen der wichtigsten Teilprozesse der Produktentwicklung, die Berechnung der im Erzeugnis verbauten Maschinenelemente. Deren Auslegung, Optimierung und Nachrechnung sind einerseits Teil eines übergeordneten und arbeitsteiligen Entwicklungsprozesses, bedürfen aber zur eigenen Bestimmung ebenfalls vieler einzelner parallel ablaufender interdisziplinärer Aktivitäten.

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung einer Methode zur Systematisierung und Erhöhung der Transparenz der Abläufe in der Produktentwicklung mit dem Schwerpunkt der Auslegung und Berechnung von Maschinenelementen und deren Baugruppen. Es wurden die Baugruppen- und Prozessbildung, die Werkzeuge für die Modellbildung sowie eine prototypisch realisierte Baugruppenberechnung vorgestellt.

Auf Basis der durchgeführten Literaturrecherche wurden die grundlegenden Vorgehensweisen und Methoden zur Organisation und Durchführung von Berechnungsprozessen identifiziert, analysiert und auf die Anwendbarkeit der Lösung, der im Rahmen dieser Arbeit gestellten Aufgaben zur Systematisierung und Erhöhung der Transparenz der Berechnungsabläufe im Bereich der Maschinenelementeberechnungen untersucht.

Die Grundlage der neuen Methode bilden Ansätze, die sich bereits in der Konstruktionsmethodik und in der Informationstechnik bewährt haben. Ergänzt mit den Konzepten der Klassifikation und Prozessautomatisierung sowie den mathematisch-technischen Berechnungsalgorithmen der Ingenieurwissenschaften bilden sie die Basis dieser Methode. Die für die Automatisierung notwendige Formalisierung der Methode wurde basierend auf der Theorie der gerichteten Graphen, der SysML (Systems Modeling Language) und des Business Process Management (BPM) Ansatzes durchgeführt. Die gewonnenen theoretischen Erkenntnisse wurden in Form eines prototypisch entwickelten Design Process Management Systems realisiert, welches die Erreichung der in der Arbeit gesetzten Ziele ermöglicht und für ein nachhaltiges Wissensmanagement bei der Organisation der Berechnungsabläufe sorgt. Die Besonderheiten dieses workfloworientierten Management Systems liegen einerseits in einem Wissens-Repository, welches die modularen Bestandteile zur Definition der Berechnungsabläufe beinhaltet sowie andererseits in einem interaktiven Prozess-Designer, mit dessen Hilfe die Abläufe graphisch, schnell und intuitiv modelliert und ausgeführt werden können.



In den Produktplanungsphasen vom Anforderungsmanagement bis zur Validierung nach dem Prinzip des „Systems Engineering“ kann dabei auf eine Bibliothek modularer Berechnungsobjekte in Form von Prozess-Bausteinen zugegriffen werden, deren Schnittstellen und Datenstrukturen ausnahmslos einheitlichen Definitionen entsprechen. Gemeinsam mit der nach eCl@ss - Standard entwickelten Merkmalsstruktur der Berechnungsobjekte wird so eine hohe Wiederverwendbarkeit erzielt. Die Klassifizierung der Berechnungsobjekte orientiert sich an konstruktionssystematischen Gesichtspunkten.

Der eigentliche Prozess der Baugruppenbildung erfolgt mit einem Editor, der die Berechnungsobjekte miteinander verknüpft, indem er die zugehörigen Informationsobjekte, Datenbankzugriffe und Algorithmen in die Benutzeroberfläche und in den Prozess einbindet. Ebenso können externe Prozesse und Datenquellen nahtlos in die Modellierung einfließen. Schließlich wird die Vorgehensweise bei der Modellbildung einer Berechnungs-Baugruppe am Beispiel einer Getriebeauslegung dargestellt. Ausgehend von den geforderten Merkmalen der Lösungskonfiguration wird die Funktionsstruktur durch analoge Berechnungsobjekte abgebildet und anschließend über eine geeignete Logik miteinander verknüpft. Der dabei entstehende Prozess bildet die neue Berechnungs-Baugruppe der Getriebeauslegung und kann schließlich für Parameterstudien herangezogen werden.

Mit den prozessgebundenen Berechnungsbaugruppen wurde eine Methode vorgestellt, die den Konstrukteuren die Möglichkeit bietet, ihre zum Teil mehrstufigen und rekursiven Berechnungen durch den Einsatz modularer Funktionsbausteine verbindlich, nachvollziehbar und vor allem zeitsparender zu gestalten. Dabei sind die Arbeitsergebnisse allen Ingenieuren und Abteilungen einer Community, vor allem auch dem Qualitätsmanagement und damit einem kontinuierlichen Verbesserungsprozess zugänglich. Inwieweit die hier vorgetragenen Methoden auch den Validierungsprozess unterstützen können, bleibt weiteren Untersuchungen vorbehalten.

## 5.2 Ausblick

Eine wichtige Voraussetzung für die Weiterentwicklung der Methode „Prozessgebundene Berechnungsbaugruppen“ ist der Ausbau des Wissens-Repository um weitere Merkmale und Funktionsbausteine sowie Teilprozesse. Dabei müssen auch andere Fachgebiete wie Physik, Werkstoffwissenschaft, Chemie einbezogen werden, um letztendlich den kompletten Entwicklungsprozess abbilden zu können. Im Rahmen der Arbeit wurden nur die grundlegenden Elemente zur Beschreibung der Berechnungsprozesse formalisiert. Weitere Tätigkeitsfelder sind die Erweiterung der Modellierungssprache um Aspekte der Massenverarbeitung von Daten und Schnittstellenelemente zu den gängigsten CAD, FEM sowie Daten Management Systemen. Eine Erweiterung auf Cloud-Applikation würde sich anbieten, um das Wissen in dem aufgebauten Repository vielen zugänglich zu machen.

Die Langzeitaufbewahrung des Contents (Funktionsbausteine) erfolgt derzeit mit Hilfe der MathML Sprache. Das sichert eine Programmiersprachenunabhängige Aufbewahrung des Contents. Um den Less-Code Ansatz weiterhin zu unterstützen ist es sinnvoll die MathML Sprache nicht nur zur Visualisierung und Dokumentation der im Funktionsbaustein implementierten Funktion zu nutzen, sondern auch für deren Eingabe. Das erhöht die Benutzerfreundlichkeit und minimiert die Fehler bei der Definition des Funktionsbausteines denn die mathematischen Ausdrücke sind damit deutlich verständlicher als in jeglicher Programmiersprache.

Die Entwicklung einer Suchfunktion auf Basis der für alle Merkmale und Funktionsbausteine eingetragenen Synonyme kann das Wiederfinden des gewünschten Funktionsbausteines erhöhen. Um Abhängigkeiten innerhalb der prozessgebundenen Berechnungs-Baugruppen sichtbar werden zu lassen, die wegen der Komplexität des Systemverhaltens oft verborgen bleiben, wäre es sinnvoll darüber hinaus eine Methode zu entwickeln, mit der auf der Basis von Design Struktur Matrizen der Einfluss einzelner Objekte aufeinander analysiert werden kann.

## 6 Literaturverzeichnis

1. Abramovici, M. und Schulte, S.: Logische Fortsetzung der PDM-Ansätze oder Neuauflage des CIM-Debakels; VDI-Berichte, Vol. 1819. I 2 P 2004, VDI, Düsseldorf, 2004
1. Abulawi, J.: Ansatz zur Beherrschung der Komplexität von vernetzten 3D-Cad-Modellen, Dissertation, Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg, Hamburg, 2012
2. Acél, P.: Systems-Engineering und Simulation Leitfaden zum Vorgehen; 1996, <https://www.acel.ch/pdf/simulation-nach-systems-engineering-pb002.pdf>, Abgerufen am 19.10.2016
3. Aigner, M.: Graphentheorie, Eine Einführung aus dem 4-Farben Problem; Springer Verlag, Wiesbaden, 2015
4. Alt, O.: Modellbasierte Systementwicklung mit SysML – In der Praxis; Carl Hanser Verlag, München, 2012
5. Bartsch, C.: Modellierung und Simulation von IT-Dienstleistungsprozessen; Dissertation, Karlsruher Institut für Technologie, 2010
6. Baumberger, G. C.: Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten; Dissertation, TU München, 2007
7. Berkau, C.: Instrumente der Datenverarbeitung für das effiziente Prozesscontrolling, Kostenrechnungspraxis 2 (Sonderheft 2), 1998, S. 27-32
8. Blessing, U. C.: GETRAG HybridDrive 7HDT300 The Optimized Hybridization; 2012, [http://www.getrag.com/media/media/text/dtf\\_2012/DTF\\_2012\\_05\\_Hybrid\\_Drive.pdf](http://www.getrag.com/media/media/text/dtf_2012/DTF_2012_05_Hybrid_Drive.pdf), Abgerufen am 13.12.2016
9. BMI: Handbuch für Organisationsuntersuchungen und Personalbedarfsermittlung;: Bundesministerium des Innern / Bundesverwaltungsamt, Berlin Köln, 2016
10. Bösing, K. D. und Geldmacher, J.: Transformation einer EPK in eine formalisierte Methode; Wissenschaftliche Beiträge 2014, 2014, S. 95-101
11. Büchner, T.: Introspektive modellgetriebene Softwareentwicklung; Dissertation, TU München, 2007
12. Bulligk, C. K.: Theoretische Untersuchungen zur modularisierten Berechnung und Auslegung von Getrieben; Diplomarbeit, TU Dresden, 2009
13. Bullinger, H. J., Spath, D., Warnecke, H. J., Westkämper (Hrsg)., E.: Handbuch Unternehmensorganisation, Strategien, Planung, Umsetzung; Springer, Berlin Heidelberg, 2009

14. Conrad, K. J.: Grundlagen der Konstruktionslehre; Carl Hanser Verlag, München, 2013
15. Daenzer, W. F. und Huber, F.: Systems Engineering. Industrielle Organisation, Zürich, 1999
16. Daenzer, W. F. und Huber, F.: Systems Engineering, Methodik und Praxis; Industrielle Organisation, Zürich, 2002
17. Diehl, H.: Systemorientierte Visualisierung disziplinübergreifender Entwicklungsabhängigkeiten mechatronischer Automobilsysteme, Dissertation. TU München, 2009
18. DIN 1301-1:2010-10: Einheiten - Teil 1: Einheitenamen, Einheitenzeichen; Beuth, Berlin
19. DIN 281:2010-10: Wälzlager - Dynamische Tragzahlen und nominelle Lebensdauer; Beuth, Berlin
20. DIN 3960:1987-03: Begriffe und Bestimmungsgrößen für Stirnräder (Zylinderräder) und Stirnradpaare (Zylinderradpaare) mit Evolventenverzahnung, Beuth, Berlin
21. DIN 3990-2:1987-12: Tragfähigkeitsberechnung von Stirnrädern, Berechnung der Grübchentragfähigkeit; Beuth, Berlin
22. DIN 3990-3:1987-12: Tragfähigkeitsberechnung von Stirnrädern, Berechnung der Zahnfußtragfähigkeit; Beuth, Berlin
23. DIN 4000-141:2016-10: Sachmerkmal-Listen – Teil 141: Wälzfräser mit Bohrung oder Schaft für Evolventenverzahnung, außen; Beuth, Berlin
24. DIN 4002-2:2007-07: Merkmale und Geltungsbereiche zum Produktdatenaustausch – Teil 2: Begriffe und konzeptionelles Informationsmodell; Beuth, Berlin
25. DIN 4002-3:2006-3: Merkmale und Geltungsbereiche zum Produktdatenaustausch - Teil 3: Inhaltliche und datentechnische Definition der Attribute für Strukturelemente; Beuth, Berlin
26. DIN 743-1:2012-12: Tragfähigkeitsberechnung von Wellen und Achsen; Beuth, Berlin
27. DIN 780-1:1977-05; Modulreihe für Zahnräder, Moduln für Stirnräder; Beuth, Berlin
28. DIN EN 61360:2004-12: Genormte Datenelementtypen mit Klassifikationsschema für elektrische Bauteile - Teil 1: Definitionen - Regeln und Methoden; Beuth, Berlin
29. DIN EN ISO 9000:2015-11: Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2015), Beuth, Berlin
30. DIN EN ISO 9001:2015-11: Qualitätsmanagementsysteme – Anforderungen (ISO 9001:2015); Beuth, Berlin

31. DIN IEC 60050-351:2014-09: Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik; Beuth, Berlin
32. DIN ISO 21771:2014-08: Zahnräder - Zylinderräder und Zylinderradpaare mit Evolventenverzahnung - Begriffe und Geometrie (ISO 21771: 2007); Beuth, Berlin
33. DIN: Standardisierung des ISO-XML-Formats für Normen; Beitrag in DIN-Mitteilungen 2016-07; Beuth, Berlin
34. Düsing, C.: Produktentwicklung auf der Basis von Systems Engineering; Mitteilungen aus dem Institut für Maschinenwesen der TU Clausthal, Band 26, 2001, S. 67-70.
35. Dyla, A.: Modell einer durchgängig rechnerbasierten Produktentwicklung, Dissertation, TU München, 2002
36. Ebert, T.: Die Systematisierung visueller Darstellungsformen in der sozialwissenschaftlichen Forschung; Dissertation, Philipps-Universität Marburg, 2013
37. eCl@ss e.V.: Whitepaper Industrie 4.0., o. D.,  
[http://www.eclass.eu/static/documents/wiki/Whitepaper/ecl-Whitepaper-Industrie40\\_DE.pdf](http://www.eclass.eu/static/documents/wiki/Whitepaper/ecl-Whitepaper-Industrie40_DE.pdf),  
Abgerufen am 20.10.2016
38. eCl@ss e.V.: eCl@ss classification and produkt deskription, 2016,  
[http://wiki.eclass.eu/wiki/Main\\_Page](http://wiki.eclass.eu/wiki/Main_Page), Abgerufen am 14.12.2016
39. Ehrlenspiel, K.: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produkterstellung und Konstruktion; Carl Hanser Verlag, München Wien, 1995
40. Ehrlenspiel, K.: Integrierte Produktentwicklung. Methoden für Prozessorganisation, Produkterstellung und Konstruktion; Carl Hanser Verlag, München Wien, 2009
41. Eigner, M., Gilz, T., Zafirov, R.: Interdisziplinäre Produktentwicklung: Modellbasiertes Systems Engineering, 2012 <http://www.plmportal.org/de/forschung-detail/interdisziplinaere-produktentwicklung-modellbasiertes-systems-engineering.html>  
Abgerufen am 19.10.2016
42. Eigner, M., Gilz, T., Zafirov, R: Funktionale Produktbeschreibung, Modellbasiertes Systems Engineering; IT&PRODUCTION (09) 2011
43. Eigner, M., Roubanov, D., Zafirov, R.: Modellbasierte virtuelle Produktentwicklung; Springer Vieweg Verlag, Berlin Heidelberg, 2014
44. Eigner, M. und Schleidt, T.: Unternehmensübergreifende Zusammenarbeit im Cross Enterprise Engineering; wt Werkstatttechnik, Springer-VDI-Verlag, Düsseldorf, 2006

45. Eigner, M. und Stelzer, R.: Product Lifecycle Management; Springer Verlag, Berlin Heidelberg, 2009
46. Endres, A. und Rombach, D.: A Handbook of Software and Systems Engineering; Pearson/Addison-Wesley, Harlow, 2003
47. ESTECO SpA: Integration & Processautomation, modeFRONTIER, o.D.,  
<http://www.esteco.com/modefrontier/integration-process-automation>  
Abgerufen am 14.12.2016
48. Feldhusen, J. und Grote, K. H.: Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung; Springer Vieweg Verlag, Berlin Heidelberg, 2013
49. Frei, N.: Der Konstruktionsprozess: Wirkmodelle basierend auf Constraints; Dissertation, Eidgenössische technische Hochschule Zürich, 2002
50. Frenzel, M.: Zum Aufbau, zur Arbeitsweise und Realisierung von problemorientierten, selbstorganisierenden Programmsystemen; Dissertation, TU Dresden, 1980
51. Freund, J. und Götzer, K.: Vom Geschäftsprozess zum Workflow: Ein Leitfaden für die Praxis; Carl Hanser Verlag, München, 2008
52. Gadatsch, A.: Workflow-Management; o.D. <http://www.org-portal.org/fileadmin/media/legacy/prof. dr. a. gadatsch workflow-management.pdf>  
Abgerufen am 20.10.2016
53. Gadatsch, A., Grundkurs Geschäftsprozess-Management, Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker; Springer Vieweg Verlag, Wiesbaden, 2012
54. Gausemeier, J. et al.: Systems Engineering in der industriellen Praxis; Paderborn: Heinz Nixdorf Institut (Universität Paderborn), Fraunhofer-Institut für Produktionstechnologie, UNITY AG., 2013
55. Gausemeier, J., Ebbesmayer, P., Kallmeyer, F.: Produktinnovation, Hanser Verlag, München, 2001
56. Gausemeier, J., Trächtler, A., Schäfer, W.: Semantische Technologien im Entwurf mechatronischer Systeme: Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten; Carl Hanser Verlag, München, 2014
57. Gehring, H. und Gadatsch, A.: Ein Rahmenkonzept für die Modellierung von Geschäftsprozessen und Workflows; Fachbereichsbericht Nr. 274, FernUniversität Hagen, 1999
58. Götz, M.: BPMN 2.0 Tutorial - Kompakte Einführung in die BPMN 2.0; iTransparent GmbH | Business Process Architects, 2011

59. Grätz, U. und Rodig, F.: Requirement meets Validation; Vortrag anlässlich ProSTEP iViP Symposium 2016, Stuttgart, 2016
60. Gründer, W., 2010. Holistic Approach in Automotive and Mechanical Design requires Structured but Flexible Workflows, Proceedings of CONAT 2010, Brasov, 2010
61. Gründer, W. und Polyakov, D.: Design Process Management - Eine Methode zur integrativen bearbeitung intellektueller und automatisierter Vorgänge in der Produktentwicklung; Dresdener Maschinenelemente-Kolloquium - DMK 2011, TUDpress, Dresden, 2011, S. 133-144
62. Haasis, S.: Digital Transformation and the future with Model Based Systems Engineering; Vortrag anlässlich ProSTEP iViP Symposium 2016, Stuttgart, 2016
63. Hammer, M. und Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution; Harper Business, New York, 1988
64. Haun, M.: Handbuch Robotik: Programmieren und Einsatz intelligenter Roboter; Springer Vieweg Verlag, Berlin Heidelberg, 2013
65. Helming, A.: Funktionale vs. Prozessorientierte Organisation; Das IPL - Magazin, Issue 34, 2016
66. Henning, P. A. und Vogelsang, H.: Handbuch Programmiersprachen: Softwareentwicklung zum Lernen und Nachschlagen; Carl Hanser Verlag, München, 2006
67. Herbst, D.: Erfolgsfaktor Wissensmanagement; Cornelsen Verlag, Berlin, 2000
68. Hering, E., Gutekunst, J., Dyllong, U.: Handbuch der praktischen und technischen Informatik; Springer Verlag, Berlin, 2000
69. Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web; Springer Verlag, Berlin Heidelberg, 2007
70. Hitz, M., Kappel, G., Kapsammer, E., Retschitzegger, W.: UML @ Work. Objektorientierte Modellierung mit UML 2; Dpunkt-Verlag, Heidelberg, 2005
71. Hubka, V., Theorie der Konstruktionsprozesse, Springer Verlag, Berlin Heidelberg, 1976
72. Institut der deutschen Wirtschaft Köln: Klassifikationsstandards auswählen und einsetzen: Handlungsempfehlung zum Einsatz von eBusiness-Standards; Prozeus, Köln, 2010
73. Institut für Technologie und Arbeit (ITA): Entwicklungsprozess: Vorgehensmodell nach VDI 2221; 2013, <http://www.optimus-spitzencluster.de/entwicklungsprozessvorgehensmodellnachvdi.pdf>, Abgerufen am 19.10.2016

74. ISO 13584-1:2001-04; Industrielle Automatisierungssysteme und Integration - Teilebibliothek - Teil 1: Überblick und Grundlagen; ISO, Switzerland
75. ISO 6336-1:2006-09; Calculation of load capacity of spur and helical gears - Part 1: Basic principles, introduction and general influence factors; ISO, Switzerland
76. ISO 6336-2:2006-09; Calculation of load capacity of spur and helical gears - Part 2: Calculation of surface durability (pitting); ISO, Switzerland
77. ISO/IEC 40314:2016-03; Information technology - Mathematical Markup Language (MathML) Version 3.0 2nd Edition; ISO, Switzerland
78. Jaroš, M.: Integration des STEP-Produktmodells in den Getriebeentwicklungsprozess; Dissertation; TU München, 2007
79. JAXenter: Hält die SysML was sie verspricht?; 2008, <https://jaxenter.de/halt-die-sysml-was-sie-verspricht-2-9250>, Abgerufen am 13.12.2016
80. Jung, A.: Studie zum Knowledge Management und zur Informationsbeschaffung von Ingenieuren (Ingenieure der Elektro- und Nachrichtentechnik in Großunternehmen); Diplomarbeit, FH Köln, 2001
81. Kaiser, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme; Dissertation, Universität Paderborn, 2013
82. Kiffe, G. und Bock, T., 2013; Standardisierte Entwicklungsumgebung für die Softwareeigenentwicklung bei Audi; [https://www.dspace.com/shared/data/pdf/ankon2013/Tag2\\_PDF/2\\_Audi\\_Kiffe-Gerhard\\_Bock-Thomas.pdf](https://www.dspace.com/shared/data/pdf/ankon2013/Tag2_PDF/2_Audi_Kiffe-Gerhard_Bock-Thomas.pdf), Abgerufen am 20.10.2016
83. Koller, R.: Konstruktionslehre für den Maschinenbau: Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen; Springer Verlag, Berlin, 1994
84. Kopytkov, O.: Systemmodellierung mit SysML; 2009, <https://wiki.enterpriselab.ch/edu/workspace:modules:swe:s08:okopytkov:arbeit>, Abgerufen am 13.12.2016
85. Kühne, S.: Modellgetriebene Entwicklung im Kontext des Integration Engineering; 2007, <http://lips.informatik.uni-leipzig.de/files/Khne2007ModellgetriebeneEntwicklungimKontextdesIntegration.pdf>, Abgerufen am 19.10.2016
86. Leemhuis, H.: Funktionsgetriebene Konstruktion als Grundlage verbesserter Produktentwicklung; Dissertation, TU Berlin, 2005



87. Lindemann, U.: Methodische Entwicklung technischer Produkte; Springer, Berlin, 2009
88. Linke, H.: Stirnradverzahnung, Berechnung - Werkstoffe – Fertigung; Hanser Verlag, München Wien, 2012
89. List, H.: Development, Simulation and Test of Powertrain Systems; Virtual Vehicle 21, 2015
90. Lütke-meier, B. und Thöne, S.: Prozessorientierte Integration von Softwarekomponenten durch XML-basierte Workflow-Modelle; Diplomarbeit, Universität Paderborn, 2001
91. M.-V.: Dienstleistungsportal Mecklenburg-Vorpommern, Ausgewählte Modellierungs-Notationen im Überblick; 2011, [http://www.service.m-v.de/cms/DLP\\_prod/DLP/Servicebereich/Kooperatives\\_E-Government/\\_/01\\_Anlage\\_Modellierungsnotationen\\_im\\_Ueberblick\\_V03.pdf](http://www.service.m-v.de/cms/DLP_prod/DLP/Servicebereich/Kooperatives_E-Government/_/01_Anlage_Modellierungsnotationen_im_Ueberblick_V03.pdf)  
Abgerufen am 14.12.2016
92. Mashchenko, A.: Entwicklung eines digitalen Werkzeugs; Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden, 2015
93. Maune, G.: Möglichkeiten des Komplexitätsmanagements für Automobilhersteller auf Basis IT-gestützter durchgängiger Systeme; Dissertation, Universität-GH-Paderborn, 2002
94. Miehle, A.: Das DIN-Merkmallexikon oder warum die Industrie genormte Merkmale benötigt; IMW - Institutsmitteilung Nr. 33, 2008 S. 49-60
95. Möhringer, S.: Die neue Richtlinie VDI 2206: Entwicklungsmethodik für mechatronische Systeme; 2003,  
<http://mechatronics-net.de/gerworkgroup/sitzung6/moehringer-vdi.pdf>, Abgerufen am 19.10.2016
96. Müller, B.: Wissen managen in formal organisierten Sozialsystemen; Gabler, Wiesbaden, 2009
97. Niemann, G. und Winter, H.: Maschinenelemente Band II: Getriebe allgemein, Zahnradgetriebe – Grundlagen, Stirnradgetriebe; Springer Verlag, Berlin Heidelberg New York, 1983
98. Niemann, G. und Winter, H.: Maschinenelemente, Band 2: Getriebe allgemein, Zahnradgetriebe - Grundlagen, Stirnradgetriebe; Springer Verlag, Berlin Heidelberg, 2005
99. OMG: Object Management Group (OMG) - Business Process Model And Notation™ (BPMN™) Version 2.0.; 2011, <http://www.omg.org/spec/BPMN/2.0/>, Abgerufen am 14.12.2016
100. OMG: Object Management Group (OMG) - Object Management Group, Systems Modeling Language (OMG SysML™), Version 1.4.; 2015a, <http://www.omg.org/spec/SysML/1.4/PDF>, Abgerufen am 13.12.2016

101. OMG: Object Management Group (OMG) - Documents Associated With Unified Modeling Language™ (UML®) Version 2.5.; 2015b, <http://www.omg.org/spec/UML/2.5/>, Abgerufen am 03.05.2017
102. Oestereich, B.: Objektorientierte Geschäftsprozessmodellierung und modellgetriebene Softwareentwicklung; 2005, <http://docplayer.org/4539410-Eine-begrenzte-geschaeftsprozessmodellierung-und-modellgetriebene-softwareentwicklung.html>, Abgerufen am 19.10.2016
103. oose.: Modellbasiertes Requirements Engineering; 2016, <http://www.oose.de/training/mbre-modellbasiertes-requirements-engineering/>, Abgerufen am 13.12.2016
104. oose.: Was ist Systemsengineering; o.D., <http://www.oose.de/nuetzliches/was-ist-systems-engineering/>, Abgerufen am 14.12.2016
105. Pahl, G. und Beitz, W.: Konstruktionslehre, Methoden und Anwendung; Springer, Berlin Heidelberg, 1997
106. Panreck, K.: Methodik der Modellbasierten Systementwicklung; 2014, <https://www.fh-bielefeld.de/multimedia/...und.../Vortrag+Panreck+Solutions.pdf>, Abgerufen am 14.12.2016
107. Parlow, J., Otto, M. und Stahl, K.: Vom Lastenheft zur Verzahnung: anwendungsflexible Dimensionierung von Zahnradgetrieben mittels expliziten Entwurfsmodell; Konstruktion (3), 2016, S. 64-73
108. Partsch, H.: Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme; Springer Verlag, Berlin Heidelberg, 2010
109. Patzak, G.: Systemtechnik, Planung komplexer innovativer Systeme – Grundlagen, Methoden, Techniken; Springer Verlag, Berlin Heidelberg New York, 1982
110. Petkov, V.: Automatic Performance Engineering Workflows for High Performance Computing; Dissertation, TU München, 2014
111. Phoenix integration: Phoenix Modelcenter Integrate; 2016, <http://www.phoenix-int.com/modelcenter/integrate.php>, Abgerufen am 14.12.2016
112. Polyakov, D. und Gründer, W.: Nachhaltiges Wissensmanagement in der Konstruktion durch die prozessorientierte Herangehensweise und systematische Gestaltung von Berechnungsprozessen; Ein Beitrag anlässlich Dresdener Maschinenelemente-Kolloquium - DMK 2015, TUDpress, 2015

113. Polyakov, D. und Gründer, W.: Design Process Management – ein systematischer Ansatz zur Durchführung komplexer Maschinenelementeberechnungen; ein Beitrag anlässlich Dresdener Maschinenelemente-Kolloquium - DMK 2013, TUDpress, Dresden, 2013 S. 303-314
114. Polyakov, D. und Gründer, W.: Design Process Management; ein Beitrag anlässlich Konferenz Entwickeln – Entwerfen – Erleben EEE2012, TUDpress, Dresden, 2012
115. Polyakov, D. und Gründer, W.: Überlegungen zur digitalen Verknüpfung von Normen und standardisierten Algorithmen zu Berechnungsbaugruppen; ein Beitrag anlässlich Konferenz Entwickeln – Entwerfen – Erleben EEE2014, TUDpress, Dresden, 2014, S. 605-617
116. Polyakov, D. und Gründer, W.: Prozessgebundene Berechnungsbaugruppen: Ein Ansatz zur Lösung komplexer Entscheidungs- und Berechnungsabläufe; ein Beitrag anlässlich Konferenz Entwickeln – Entwerfen – Erleben EEE2016, TUDpress, Dresden, 2016
117. Ponn, J. und Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte: Optimierte Produkte - systematisch von Anforderungen zu Konzepten; Springer Verlag, Berlin Heidelberg, 2008
118. Ponn, J. und Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte: Systematisch von Anforderungen zu Konzepten und Gestaltlösungen; Springer Verlag, Berlin Heidelberg, 2011
119. Riggert, W.: Workflow; Bildmaterial zur Vorlesung, o. D., <http://www.wi.fh-flensburg.de/fileadmin/dozenten/Riggert/bildmaterial/Dokumentenmanagement/3-Management-WF.pdf>, Abgerufen am 14.12.2016
120. Römhild, I.: Auslegung mehrstufiger Strirnradgetriebe - Übersetzungsaufteilung für minimale Masse und Wahl der Profilverschiebung auf der Basis neuer Berechnungsgrundlagen; Dissertation; TU Dresden, 1993
121. Roth, K.: Konstruieren mit Konstruktionskatalogen; Springer Verlag, Berlin, 1994
122. Rumpe, B.: Modellierung mit UML; Springer Verlag, Berlin Heidelberg, 2011
123. Rupp, C., Queins, S., die SOPHISTen: UML 2 glasklar: Praxiswissen für die UML-Modellierung; Carl Hanser Verlag, München, 2012
124. Rüppel, U.: Vernetzt-kooperative Planungsprozesse im konstruktiven Ingenieurbau; Springer Verlag, Berlin Heidelberg, 2007
125. Scheer, A.-W.: ARIS – vom Geschäftsprozeß zum Anwendungssystem; Springer Verlag, Berlin, 2002

126. Schlecht, B.: Maschinenelemente 2: Getriebe, Verzahnungen und Lagerungen; Pearson Studium, München, 2009
127. Schlecht, B., Schulze, T., Hartmann-Gerlach, C.: Auslegung & Optimierung von mehrstufigen; o.D., [http://www.driveconcepts.com/info/paper\\_gearbox\\_de.pdf](http://www.driveconcepts.com/info/paper_gearbox_de.pdf), Abgerufen am 11.11.2016
128. Schlingloff, H.: 35. Algorithmus der Woche, Zyklensuche; 2006, <http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo35.php>, Abgerufen am 14.12.2016
129. Schwill, A.: Beschreibung paralleler Abläufe mit Petri-Netzen; o.D., <http://www.informatikdidaktik.de/HyFISCH/Informieren/Nebenlaeufigkeit/PetriNetze.pdf>, Abgerufen am 19.10.2016
130. Siegel, J.: BPMN unter der Lupe; 2008, [http://www.sigs-datacom.de/uploads/tx\\_mwjournals/pdf/OMG\\_OS\\_01\\_08.pdf](http://www.sigs-datacom.de/uploads/tx_mwjournals/pdf/OMG_OS_01_08.pdf), Abgerufen am 19.10.2016
131. Signavio: BPMN 2.0: Prozessautomatisierung mit jBPM und Activiti; 2012, <http://www.signavio.com/de/news/bpmn-2-0-prozessautomatisierung-mit-jbpm-und-activiti/>, Abgerufen am 19.10.2016
132. Spahni, D.: Problemlösungsmethodik und Projektmanagement; 2000, <http://www.spahni.com/data/uploaded/downloads/SE.pdf>, Abgerufen am 19.10.2016
133. Stachowiak, H.: Allgemeine Modelltheorie; Springer Verlag, Wien NewYork, 1973
134. Stahl, T.: Völter, M., Efttinge, S., Haase, A.: Modellgetriebene Softwareentwicklung, Techniken, Engineering, Management; dpunkt Verlag, Heidelberg, 2007
135. Steinmeier, E.: Realisierung eines systemtechnischen Produktmodells - Einsatz in der PKW-Entwicklung; Shaker, Maastricht Herzogenrath, 1999
136. Störrle, H.: UML 2 für Studenten; Pearson Studium, München, 2005
137. Uiterwyk, H.: Klassifikationssysteme im Engineering; 2012, [http://www.industry-forum.biz/tl\\_files/iforum/review/2012/pdf/eCLASS\\_Uiterwyk\\_Poppinghuys\\_DE.pdf](http://www.industry-forum.biz/tl_files/iforum/review/2012/pdf/eCLASS_Uiterwyk_Poppinghuys_DE.pdf), Abgerufen am 20.10.2016
138. van der Aalst, W. M., ter Hofstede, A. H. M., Weske, M.: Business process management: A Survey; International Conference, BPM 2003 Eindhoven, The Netherlands, Springer Verlag, Berlin Heidelberg, 2003
139. VDI 2206: 2004; Entwicklungsmethodik für mechatronische Systeme; VDI, Düsseldorf

140. VDI 2221:1993; Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte; VDI, Düsseldorf
141. VDI 2223:2004; Methodisches Entwerfen technischer Produkte; VDI, Düsseldorf
142. Vorest AG: Prozessmanagement; 2015, <http://prozessmanagement.me/prozessmanagement/>, Abgerufen am 03.05.2017
143. W3C: World Wide Web Consortium – W3C, Extensible Markup Language (XML) 1.1 (Second Edition); 2006, <https://www.w3.org/TR/2006/REC-xml11-20060816/>, Abgerufen am 13.12.2016
144. W3C: World Wide Web Consortium – W3C, HTML5, A vocabulary and associated APIs for HTML and XHTML; 2014a, <https://www.w3.org/TR/html5/>, Abgerufen am 14.12.2016
145. W3C: World Wide Web Consortium – W3C, Mathematical Markup Language (MathML) Version 3.0 2nd Edition, 2014b, <https://www.w3.org/TR/MathML/>, Abgerufen am 13.12.2016
146. Weilkiens, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design; dPunkt Verlag, Heidelberg, 2006
147. Weilkiens, T.: Systems Engineering mit SysML, Konferenz Software meets Business; [http://www.sigs.de/download/oop\\_07/Weilkiens%20Ndo3%20%20%20%20WR.pdf](http://www.sigs.de/download/oop_07/Weilkiens%20Ndo3%20%20%20%20WR.pdf), Abgerufen am 13.12.2016, 2007
148. Weishaupt, B.: Interdisziplinärer prozessorientierter Entwurf technischer Systeme; Dissertation, Herbert Utz Verlag, München, 1998
149. Westkämper, E.: Einführung in die Organisation der Produktion; Springer Verlag, Heidelberg, 2006
150. Wittel, H., Muhs, D., Jannasch, D., Voßiek, J.: Roloff / Matek Maschinenelemente, Normung, Berechnung, Gestaltung; Vieweg Teubner Verlag, Wiesbaden, 2011
151. Wörzberger, R.: Management dynamischer Geschäftsprozesse auf Basis statischer Prozessmanagementsysteme; Dissertation, Technische Hochschule Aachen, 2010
152. Xiang, L.: Entwicklung von Modellen generischer Managementprozesse für die Gestaltung und Lenkung prozessorientierter Unternehmen; University of Bamberg Press, Bamberg, 2015
153. Zingel, J. C.: Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des ZHO-Prinzips; Dissertation, Karlsruher Institut für Technologie, 2013

## 7 Abbildungsverzeichnis

Abbildung 1: Funktionsorientierte vs. prozessorientierte Produktentwicklung.....	4
Abbildung 2: Schematischer Ablauf einer Berechnung .....	7
Abbildung 3: Allgemeiner Ablauf der Fußfestigkeitsberechnung nach DIN 3990:1987.....	8
Abbildung 4: Produktentwicklung in Produktlebenszyklus (Eigner und Stelzer 2009) .....	11
Abbildung 5: Wesentliche Aspekte der Produktentwicklung (Ponn und Lindemann, 2008).....	12
Abbildung 6: Produktlebenszyklus (adaptiert nach Eigner und Stelzer 2009) .....	12
Abbildung 7: Antriebsstrang mit Hybriddrive 7HDT300 der Firma GETRAG (Blessing, 2012).....	14
Abbildung 8: Komplexität nach (Steinmeier, 1999).....	14
Abbildung 9: Konstruktionsprozess von heute (Gründer, 2010).....	16
Abbildung 10: Wissensverteilung im Unternehmen (Jung, 2001).....	17
Abbildung 11: Effizienzsteigerung durch ein Wissens-Repository (Kiffe und Bock, 2013).....	18
Abbildung 12: Prinzip ModelCenter® Integrate, Phoenix Integration, Inc (Phoenix integration, 2016)20	
Abbildung 13: Prinzip von dem Tool modeFRONTIER, adaptiert nach (ESTECO SpA, o.D.).....	21
Abbildung 14: Vorgehensweise beim Konstruieren nach VDI 2221:1993.....	24
Abbildung 15: V-Modell für den Entwurf mechatronischer Systeme (Panreck, 2014) .....	25
Abbildung 16: Vereinfachte Darstellung der Konstruktion als Informationsverarbeitungs- prozess adaptiert nach (Hubka, 1976).....	28
Abbildung 17: Schematische Darstellung eines Konstruktionsschrittes (ebd.) .....	28
Abbildung 18: Konstruktionsarbeitsschritt am Beispiel "Reibkraft bestimmen" (Feldhusen und Grote, 2013) .....	29
Abbildung 19: Bestimmung der Produkteigenschaft.....	29
Abbildung 20: Semantische Kopplung des Funktionsverständnisses in der Ingenieurssprache auf die funktionszentrierte Modellbildung (Zingel, 2013).....	30
Abbildung 21: Auszug aus der Prozessorientierten Darstellung der Konstruktion (Feldhusen und Grote, 2013) .....	30
Abbildung 22: Schematische Darstellung eines Prozesses .....	33
Abbildung 23: Unterteilung der Prozesse nach (Berkau, 1998) .....	33

Abbildung 24: Veranschaulichung des Begriffes "technischer Prozess" (Hering, et al., 2000).....	34
Abbildung 25: Prozess vs. Workflow (Freund und Götzer, 2008).....	35
Abbildung 26: Elemente der Petri-Netze .....	38
Abbildung 27: Beispiel eines Petri-Netzes (Freund und Götzer, 2008).....	38
Abbildung 28: Grundelemente einer Aktivität, adaptiert nach (Rupp, et al., 2012) .....	39
Abbildung 29: Darstellung der Abarbeitung einer Aktion nach dem Token-Prinzip.....	40
Abbildung 30: Beispiel der Anwendung von Pools, adaptiert nach (Hitz, et al., 2005).....	41
Abbildung 31: Beispiel des Informationsflusses in Aktivitätsdiagrammen.....	42
Abbildung 32: EPK Beispiel (Bartsch, 2010).....	43
Abbildung 33: Grundelemente von BPMN.....	44
Abbildung 34: Beispiel eines BPMN-Prozessdiagramms (Götz, 2011) .....	44
Abbildung 35: Hierarchie durch SUB-Prozesse .....	45
Abbildung 36: Die wesentlichen Bestandteile von Systems Engineering, adaptiert nach Daenzer und Huber, 1999.....	47
Abbildung 37: Problemlösungszyklus im Systems Engineering (Acél, 1996, Daenzer und Huber, 2002) .....	48
Abbildung 38: Dokumentenbasiert vs. Modellbasiert (Eigner, et al., 2014) .....	49
Abbildung 39: Diagrammarten in SysML (Partschi, 2010).....	50
Abbildung 40: Beispiel mit Erweiterung des Aktivitätsdiagrammes (vgl. ebd.).....	51
Abbildung 41: Beispiel eines Anforderungsdiagramms (JAXenter, 2008) .....	52
Abbildung 42: Anforderungstabelle zur Abbildung 41 (JAXenter, 2008) .....	52
Abbildung 43: Beispiel der Zuweisung einer Anforderung einem Systembaustein.....	52
Abbildung 44: Beispiel eines Blockdefinitionsdiagramms (Partschi, 2010) .....	53
Abbildung 45: Beispiel eines internen Blockdiagramms (Partschi, 2010) .....	54
Abbildung 46: Zusicherungsbaustein „Pumpensteuerung“ (Partschi, 2010) .....	54
Abbildung 47 Beispiel eines Zusicherungsdiagramms (vgl. ebd.).....	55
Abbildung 48: Systementwicklung nach V-Diagramm aus MBSE (Haasis, 2016).....	57
Abbildung 49: Integration der physikalischen Simulation in Systems Engineering (Grätz und Rodig, 2016) .....	58

Abbildung 50 Modellbildung entlang des Produktentwicklungsprozesses .....	59
Abbildung 51: Modellgetriebene Konstruktion, adaptiert nach (Kühne, 2007) .....	60
Abbildung 52: Klassischer Getriebe-Berechnungsprozess .....	61
Abbildung 53: Getriebestruktur mit funktionalen Zusammenhängen (Bulligk, 2009) .....	62
Abbildung 54: Schematischer Ablauf der Getriebeauslegung .....	63
Abbildung 55: Explizites Getriebeauslegung (Parlow, et al., 2016) .....	64
Abbildung 56: Entstehung der Anforderungen (Zingel, 2013) .....	69
Abbildung 57: Technische Anforderungen .....	69
Abbildung 58: Beispiel eines Anforderungsdiagramms (Partsch, 2010) .....	71
Abbildung 59: Produkteigenschaft aus Anforderung ableiten .....	71
Abbildung 60: Lebenszyklus eines Produktes (DIN 4002-2:2007-07) .....	73
Abbildung 61: Aufbau eCl@ss (eCl@ss, 2016) .....	76
Abbildung 62: Beispiel der Klassifikation der Verzahnung nach eCl@ss-System .....	76
Abbildung 63: Berücksichtigung der Einheiten beim Berechnen .....	80
Abbildung 64: Elementare Berechnung .....	80
Abbildung 65: Graphische Darstellung der Vernetzung von Berechnungsfunktionen .....	81
Abbildung 66: Funktionsaufbau nach dem EDV-Prinzip .....	81
Abbildung 67: Einfaches SysML-Aktivitätsdiagramm mit zwei Aktionen .....	83
Abbildung 68: Beispiel der Verzahnungsberechnung mit Aktivitätsdiagramm .....	84
Abbildung 69: Dokumentenmigrationsprinzip .....	84
Abbildung 70: Prinzip des gemeinsamen (zentralen) Speichers .....	85
Abbildung 71: Datenfluss zwischen den Funktionsbausteinen .....	85
Abbildung 72: Beispielprozess .....	86
Abbildung 73: Funktionsbaustein .....	87
Abbildung 74: Hierarchische Darstellung der Aufbaustruktur .....	87
Abbildung 75: Graphische Darstellung einiger Modellierungselemente in BPMN .....	87
Abbildung 76: Verwendung von Merkmalsblöcken in eCl@ss (Uiterwyk, 2012) .....	91
Abbildung 77: Beispiel der Anwendung von Merkmalsblöcken (vgl. ebd.) .....	91



Abbildung 78: Auszug aus den Merkmalsleisten mit Merkmalsgruppen .....	92
Abbildung 79: Bezugsprofil (Basisprofil) nach DIN 4000-141:2016-10 .....	94
Abbildung 80: Schematischer Ablauf der Getriebeberechnung .....	97
Abbildung 81: Top-Down Prinzip (ebd.) .....	97
Abbildung 82: Systemarchitektur des Design Process Management Systems (Polyakov und Gründer, 2016, Mashchenko, 2015).....	100
Abbildung 83: GUI vom Prozess-Designer .....	101
Abbildung 84: Verbindung zweier Funktionsbausteine .....	102
Abbildung 85: Input- und Output-Eigenschaften eines Funktionsbausteines .....	103
Abbildung 86: Definition eines neuen Merkmals für einen Funktionsbaustein.....	103
Abbildung 87: Schema für die Definition eines neuen Merkmals .....	104
Abbildung 88: Auszug aus der Merkmalsleiste der Klasse Stirnrad.....	104
Abbildung 89: Struktur eines XML-Dokumentes .....	106
Abbildung 90: Graphische Darstellung der Berechnung des Stirnradmoduls $m_t$ .....	106
Abbildung 91: Anlegen eines Funktionsbausteines .....	108
Abbildung 92: Aufbau des Infosystems.....	109
Abbildung 93: Auszug aus der Geometrieberechnung der Verzahnung (Wittel, et al., 2011).....	110
Abbildung 94: Umsetzung des Beispielprozesses.....	111
Abbildung 95: Definition der XOR-Auswahl.....	111
Abbildung 96: Beispielprozess mit Parallel-Gateways.....	112
Abbildung 97: Beispielprozess mit bedingten Kanten .....	113
Abbildung 98: Beispiel einer Endlosschleife .....	113
Abbildung 99: Algorithmus für die Zyklensuche (Schlingloff, 2006) .....	114
Abbildung 100: Eigenschaften des Solver1D-Bausteines .....	115
Abbildung 101: Beispiel eines Teilprozesses .....	116
Abbildung 102: Beispiel der Eingabeseite des Prozesssimulators.....	117
Abbildung 103: Beispiel der Ausgabeseite des eingebauten Prozesssimulators.....	117
Abbildung 104: Sub-Prozess aufgeklappt .....	119

Abbildung 105: Architektur des entwickelten PM Systems .....	121
Abbildung 106: Ableitung der Auslegungsroutinen aus den Nachrechnungsalgorithmen (Jaroš, 2007) .....	125
Abbildung 107: Iterative Vorgehensweise der Auslegung nach (Jaroš, 2007) .....	125
Abbildung 108: Ablauf der Erstauslegung der Verzahnung (ebd.) .....	126
Abbildung 109: Ablauf der Auslegung.....	126
Abbildung 110: Modellieren mit dem Prozess-Designer .....	129
Abbildung 111: Angabe der Prozessvariablen mit dem Prozess-Editor.....	130
Abbildung 112: Ablauf der modellierten Auslegung von Stufe 1 .....	130
Abbildung 113: Ablauf der modellierten Berechnung der Flankensicherheit SH .....	131
Abbildung 114: Teilprozess für die Auslegung einer Verzahnungsstufe .....	133
Abbildung 115: Ablauf der Auslegung der beiden Getriebestufen .....	133
Abbildung 116: Automatisch erstellte Applikation für die Auslegung eines Getriebes .....	135

## 8 Tabellenverzeichnis

Tabelle 1: Gegenüberstellung Prozess und Workflow, adaptiert nach (Gadatsch, 2012).....	35
Tabelle 2: Beispiel der Anforderungen an ein Getriebe .....	71
Tabelle 3: Beispiel der Anforderungseigenschaften .....	72
Tabelle 4: Beispiel der Anforderungseigenschaften erweitert.....	72
Tabelle 5: Mögliche Datentypen eines Merkmales (vgl.: DIN 4002-3:2006-3) .....	78
Tabelle 6: Basisgrößen .....	79
Tabelle 7: Abgeleitete Größen (DIN 1301-1:2010-10).....	79
Tabelle 8: Umrechnungstabelle der Einheiten (System 1 – metrisch, System 2 – US-amerikanisch) ..	80
Tabelle 9: Funktionen für die Ermittlung des Wertes von „ad“ .....	82
Tabelle 10: Funktionen für die Ermittlung des Wertes von „a“ .....	82
Tabelle 11: Auszug aus den Merkmalslisten für Wälzfräser (vgl. ebd.).....	94
Tabelle 12: Zuordnung der Merkmalskennungen nach DIN 4000-141 zu den Kurzzeichen der DIN 3960:1987-03) .....	95
Tabelle 13: Daten des zu entwickelnden Beispielgetriebes .....	123
Tabelle 14: Abhängigkeit der Anzahl der Stufen von der Gesamtübersetzung nach (Römhild, 1993)	124
Tabelle 15: Aufteilung der Gesamtübersetzung nach (Römhild, 1993).....	124
Tabelle 16: Startwerte für die Faktoren in Gleichung (4.2), (vgl. ebd.) .....	127
Tabelle 17: Ergebnisse der Auslegung der ersten Stufe .....	132
Tabelle 18: Ergebnisse der Auslegung der zweiten Stufe .....	134

## 9 Formel- und Abkürzungsverzeichnis

### Abkürzungen & Formelzeichen

Abb.	Abbildung
act	Activity diagram
AGMA	American Gear Manufacturers Association
bdd	Blockdefinitionsdiagramm
BPM	Business Process Management
bspw.	beispielsweise
bzw.	beziehungsweise
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAX	Computer Aided X
$c_p$	Kopfspielfaktor
CPS	Cyber Physical Systems
dB	Dezibel
$d_f$	Fusskreisdurchmesser
DIN	Deutsches Institut für Normung
DPMS	Design Process Management System
HTML	Hypertext Markup Language
ibd	Internes block diagramm
ISO	Internationale Organisation für Normung
MathML	Mathematical Markup Language
MBSE	Model Based Systems Engineering
$m_n$	Normalmodul
OMG	Object Management Group
PDM	Product Data Management
PLM	Product Lifecycle Management
req	Requirements diagramm
SE	Systems Engineering
$S_{Fmin}$	Minimale Sicherheit gegen Zahnfußbruch
$S_{Hmin}$	Minimale Sicherheit gegen Grübchenbildung
SysML	Systems Modeling Language
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
z	Zähnezahl

## 10 Anhang A

Klasse Stirnrad, eCl@ss: 23-17-17-1

Liste der Merkmale

ID	Name	Symbol	Einheit	Funktion	Gruppe
1.	Stegbreite	$b_s$	mm	-	1
2.	Zahnkranzinnendurchmesser	$d_i$	mm	-	1
3.	Zähnezahl	$z$	-	-	1
4.	Zahnbreite	$b$	mm	-	1
5.	Profilverschiebung	$x$	-	-	1
6.	Kopfhöhenänderung	$k$	mm	-	1
7.	Anzahl Zahneingriffe	$n_{ZE}$	-	-	1
8.	Flankenlinienwinkelabweichung	$f_{H\beta}$	$\mu\text{m}$	-	1
9.	Teilungseinzelabweichung Stirnschnitt	$f_{pt}$	$\mu\text{m}$	-	1
10.	Profil-Formabweichung	$f_{fa}$	$\mu\text{m}$	-	1
11.	Eingriffsteilungsabweichung	$f_{pe}$	$\mu\text{m}$	-	1
12.	Verzahnungsqualität	$q_v$	-	-	1
13.	Kopfrücknahme	$c_a$	$\mu\text{m}$	-	1
14.	Teilkreisdurchmesser	$d$	mm	1	1
15.	Grundkreisdurchmesser	$d_b$	mm	2	1
16.	Betriebswälzkreisdurchmesser	$d_w$	mm	3	1
17.	Zahnkopfhöhe	$h_a$	mm	4	1
18.	Kopfkreisdurchmesser	$d_a$	mm	5	1
19.	Normalzahndicke	$s_n$	mm	6	1
20.	Zahnfußhöhe	$h_f$	mm	7	1
21.	Fußkreisdurchmesser	$d_f$	mm	8,9	1
22.	Stirnzahndicke	$s_t$	mm	10	1
23.	Fußformkreisdurchmesser	$d_{Ff}$	mm	11,12	1
24.	Kopfformkreisdurchmesser	$d_{Fa}$	mm	13,14	1
25.	Fußnutzkreisdurchmesser	$d_{Nf}$	mm	15,16	1
26.	Kopfnutzkreisdurchmesser	$d_{Na}$	mm	17,18	1
27.	Ersatzzähnezahl	$z_n$	-	19	1
28.	Zahnfußsehne	$s_{Fn}$	mm	20,21	1
29.	Zahnfußrundungsradius	$\rho_F$	mm	22	1

30.	Biegehebelarm	$h_{Fe}$	mm	23,24	1
31.	Zahnkranzdicke	$s_R$	mm	25	1
32.	Drehzahl	$n$	1/min	-	2
33.	Drehmoment	$M$	Nm	26	2
34.	Lastwechselzahl	$N_L$	-	27	2
35.	Grundkreisabweichung	$f_{pb}$	$\mu m$	28	3
36.	Einlaufbetrag	$y_p$	$\mu m$	29	3
37.	Einlaufbetrag	$y_f$	$\mu m$	30	3
38.	Hilfswert Radkörperfaktor BS	BS	-	31	3
39.	Hilfswert Radkörperfaktor SR	SR	-	32	3
40.	Bezugsprofilfaktor	$C_B$	-	33	3
41.	Resonanzdrehzahl	$n_E$	1/min	34	3
42.	Kontante Dynamikfaktor $C_{ay}$	$C_{ay}$	-	35	3
43.	Einlaufbetrag Flankenlinienwinkelabweichung	$y_\beta$	$\mu m$	36	3
44.	Einlaufbetrag Flankenlinienabweichung	$y_\alpha$	$\mu m$	37	3
45.	Zahnformfaktor	$Y_F$	-	38	4
46.	Hilfsgröße L	L	-	39	4
47.	Kerbfaktor	$q_s$	-	40	4
48.	Spannungskorrekturfaktor	$Y_S$	-	41	4
49.	Zahnkranzdickenfaktor	$Y_B$	-	42,43,44	4
50.	Hochverzahnungsfaktoren	$Y_{DT}$	-	45	4
51.	Lebensdauerfaktor	$x_{zeit}$	-	46	4
52.	Lebensdauerfaktor	$y_{zeit}$	-	47	4
53.	Lebensdauerfaktor	$Y_{NT}$	-	48,49	4
54.	Gleitschichtbreite	$\rho_-$	-	53	4
55.	Bezogenes Spannungsgefälle	$\chi_-$	-	54	4
56.	Relative Stützziffer	$Y_{\delta relT}$	-	50,51,52	4
57.	Relativer Oberflächenfaktor	$Y_{RrelT}$	-	56,57,87	4
58.	Größenfaktor	$Y_X$	-	58,59,88	4
59.	Zahnfußspannung	$\sigma_{F0}$	N/mm <sup>2</sup>	60	4
60.	Zulässige Zahnfußspannungen	$\sigma_{FP}$	N/mm <sup>2</sup>	61	4
61.	Zahnfußgrenzspannung	$\sigma_{FG}$	N/mm <sup>2</sup>	62	4
62.	Vorhandene Zahnfußspannungen	$\sigma_F$	N/mm <sup>2</sup>	63	4
63.	Zahnfußsicherheit	$S_F$	-	64	4
64.	Lebensdauerfaktor	$Z_{NT}$	-	65,66,67,68	5
65.	Relativer Krümmungsradius	$\rho$	-	69,70	5

66.	Werkstoffpaarungsfaktor	$Z_W$	-	71,72	5
67.	Zulässige Zahnflankenpressung	$\sigma_{HP}$	N/mm <sup>2</sup>	73	5
68.	Zahnflankengrenzpressung	$\sigma_{HG}$	N/mm <sup>2</sup>	74	5
69.	Zahnflankensicherheit	$S_H$	-	75	5
70.	Vorhandene Zahnflankenpressung	$\sigma_{H1}$	N/mm <sup>2</sup>	76	5
71.	Vorhandene Zahnflankenpressung	$\sigma_{H2}$	N/mm <sup>2</sup>	77	5
72.	Normalmodul	$m_n$	mm	-	1
73.	Schrägungswinkel	$\beta$	°	-	1
74.	Stützlagerabstand des Ritzels	$l$	mm	-	1
75.	Außermittigkeit des Ritzels	$s$	mm	-	1
76.	Wellendurchmesser am Ritzel	$d_{sh}$	mm	-	1
77.	Normaleingriffswinkel	$\alpha_n$	°	-	1
78.	Stirnmodul	$m_t$	mm	78	1
79.	Grundschrägungswinkel	$\beta_b$	°	79	1
80.	Stirneingriffswinkel	$\alpha_t$	°	80	1
81.	Stirngrundkreisteilung	$p_{bt}$	mm	81	1
82.	Normalgrundkreisteilung	$p_{bn}$	mm	82	1
83.	Stirneingriffsteilung	$p_{et}$	mm	83	1
84.	Normaleingriffsteilung	$p_{en}$	mm	84	1
85.	Umfangskraft am Teilzylinder	$F_t$	N	85	2
86.	Umfangsgeschwindigkeit am Teilkreis	$v$	m/s	86	2
87.	Größenfaktor	$Z_X$	-	89	5

## Liste der Funktionsbausteine

ID=1	MID=14	$d=f(z,m_n,\beta)$
$d = \frac{ z  \cdot m_n}{\cos(\beta)}$		
Quelle: DIN-ISO 21771[2014] (1)		
ID=2	MID=15	$d_b=f(z,m_n,\alpha_n,\beta)$
$d_b = \frac{ z  \cdot m_n}{\sqrt{(\tan(\alpha_n))^2 + (\cos(\beta))^2}}$		
Quelle: DIN-ISO 21771[2014] (19)		
ID=3	MID=16	$d_w=f(d_b,\alpha_{wt})$
$d_w = \frac{d_b}{\cos(\alpha_{wt})}$		
Quelle: DIN-ISO 21771[2014] (56/57)		
ID=4	MID=17	$h_a=f(m_n,h_{aP^*},x,k)$
$h_a = m_n \cdot \left( h_{aP\_Stern} + x + \frac{k}{m_n} \right)$		
Quelle: DIN-ISO 21771[2014] (36)		
ID=5	MID=18	$d_a=f(d,z,h_a)$
$d_a = d + 2 \cdot \frac{z}{ z } \cdot h_a$		
Quelle: DIN-ISO 21771[2014] (33/36)		
ID=6	MID=19	$s_n=f(m_n,x,\alpha_n)$
$s_n = m_n \cdot \left( \frac{\pi}{2} + 2 \cdot x \cdot \tan(\alpha_n) \right)$		
Quelle: DIN-ISO 21771[2014] (49)		
ID=7	MID=20	$h_f=f(h_{fP},x,m_n)$
$h_f = h_{fP} - x \cdot m_n$		
Quelle: DIN-ISO 21771[2014] (37)		



ID=8	MID=21	$d_f = f(d, h_f), \text{Außenverzahnung}$
$d_f = d - 2 \cdot h_f$		
Quelle:		

ID=9	MID=21	$d_f = f(d, h_f), \text{Innenverzahnung}$
$d_f = d + 2 \cdot h_f$		
Quelle: DIN-ISO 21771[2014] (34/37)		

ID=10	MID=22	$s_t = f(x, \alpha_n, z)$
$s_t = d \cdot \left( \frac{\pi + 4 \cdot x \cdot \tan(\alpha_n)}{2 \cdot  z } \right)$		
Quelle: DIN-ISO 21771[2014] (39)		

ID=11	MID=23	$d_{ff} = f(d, \alpha_t, h_{ap0}, m_n, x, \rho_{ap0}, \alpha_t, d_b, \alpha_0, \alpha_{wt0}, d_{Fa0}, d_{b0}), \text{Wälzfräser}$
$d_{ff} = \sqrt{\left[ d \cdot \sin(\alpha_t) - \frac{2 \cdot [h_{ap0} \cdot m_n - x \cdot m_n - \rho_{ap0} \cdot (1 - \sin(\alpha_t))]}{\sin(\alpha_t)} \right]^2 + d_b^2}$		
Quelle: DIN-ISO 21771[2014] (128/131)		

ID=12	MID=23	$d_{ff} = f(d, \alpha_t, h_{ap0}, m_n, x, \rho_{ap0}, \alpha_t, d_b, \alpha_0, \alpha_{wt0}, d_{Fa0}, d_{b0}), \text{Schneidrad}$
$d_{ff} = \sqrt{\left( 2 \cdot a_0 \cdot \sin(\alpha_{wt0}) - \sqrt{d_{Fa0}^2 - d_{b0}^2} \right)^2 + d_b^2}$		
Quelle: DIN-ISO 21771[2014] (128/131)		

ID=13	MID=24	$d_{Fa}(d_a, h_k), \text{Außenverzahnung}$
$d_{Fa} = d_a - 2 \cdot h_k$		
Quelle: DIN-ISO 21771[2014] (127)		

ID=14	MID=24	$d_{Fa}(d_a, h_k), \text{Innenverzahnung}$
$d_{Fa} = d_a + 2 \cdot h_k$		
Quelle: DIN-ISO 21771[2014] (127)		

ID=15	MID=25	$d_{NF}=f(d_{dF}, a_w, a_{wt}, d_{Fa\_GR}, d_{b\_GR}, d_b),$ Außenverzahnung
$d_{NF} = \begin{cases} d_{FF} & \text{if } d_{FF} > \sqrt{\left(2 \cdot a_w \cdot \sin(\alpha_{wt}) - \sqrt{d_{Fa\_GR}^2 - d_{b\_GR}^2}\right)^2 + d_b^2} \\ \sqrt{\left(2 \cdot a_w \cdot \sin(\alpha_{wt}) - \sqrt{d_{Fa\_GR}^2 - d_{b\_GR}^2}\right)^2 + d_b^2} & \text{otherwise} \end{cases}$		
Quelle: DIN-ISO 21771[2014] (67)		

ID=16	MID=25	$d_{NF}=f(d_{dF}, a_w, a_{wt}, d_{Fa\_GR}, d_{b\_GR}, d_b),$ Innenverzahnung
$d_{NF} = \begin{cases} d_{FF} & \text{if } d_{FF} > \sqrt{\left(2 \cdot a_w \cdot \sin(\alpha_{wt}) + \sqrt{d_{Fa\_GR}^2 - d_{b\_GR}^2}\right)^2 + d_b^2} \\ \sqrt{\left(2 \cdot a_w \cdot \sin(\alpha_{wt}) + \sqrt{d_{Fa\_GR}^2 - d_{b\_GR}^2}\right)^2 + d_b^2} & \text{otherwise} \end{cases}$		
Quelle: DIN-ISO 21771[2014] (67)		

ID=17	MID=26	$d_{Na}=f(a_w, a_{wt}, d_{FF\_GR}, d_{b\_GR}, d_b, d_{Fa}),$ Außenverzahnung
$d_{Na} = \begin{cases} \sqrt{\left(2 \cdot a_w \cdot \sin(\alpha_{wt}) - \sqrt{d_{FF\_GR}^2 - d_{b\_GR}^2}\right)^2 + d_b^2} & \text{if } d_{NF} = d_{FF} \\ d_{Fa} & \text{otherwise} \end{cases}$		
Quelle: DIN-ISO 21771[2014] (69)		

ID=18	MID=26	$d_{Na}=f(a_w, a_{wt}, d_{FF\_GR}, d_{b\_GR}, d_b, d_{Fa}),$ Innenverzahnung
$d_{Na} = \begin{cases} \sqrt{\left[2 \cdot a_w \cdot \sin(\alpha_{wt}) + \sqrt{d_{FF\_GR}^2 - (d_{b\_GR}^2)^2}\right]^2 + d_b^2} & \text{if } d_{NF} = d_{FF} \\ d_{Fa} & \text{otherwise} \end{cases}$		
Quelle: DIN-ISO 21771[2014] (69)		

ID=19	MID=27	$z_n=f(z, \beta_b, \beta)$
$z_n = \frac{z}{(\cos(\beta_b))^2 \cdot \cos(\beta)}$		
Quelle:		

ID=20	MID=28	$s_{Fn}(m_n, z_n, u, G, \rho_{fPv}),$ Außenverzahnung
$s_{Fn} = m_n \cdot \left[ z_n \cdot \sin\left(\frac{\pi}{3} - \theta\right) + \sqrt{3} \cdot \left( \frac{G}{\cos(\theta)} - \frac{\rho_{fPv}}{m_n} \right) \right]$		
Quelle: ISO 6336-T3[2006] (21)		
ID=21	MID=29	$s_{Fn}(m_n, z_n, u, G, \rho_{fPv}),$ Innenverzahnung
$s_{Fn} = m_n \cdot \left[ z_n \cdot \sin\left(\frac{\pi}{6} - \theta\right) + \left( \frac{G}{\cos(\theta)} - \frac{\rho_{fPv}}{m_n} \right) \right]$		
Quelle: ISO 6336-T3[2006] (21)		
ID=22	MID=30	$\rho_F = f(\rho_{fPv}, m_n, G, u, z_n)$
$\rho_F = \rho_{fPv} + m_n \cdot \left[ \frac{2 \cdot G^2}{\cos(\theta) \cdot (z_n \cdot \cos(\theta) \cdot \cos(\theta) - 2 \cdot G)} \right]$		
Quelle: ISO 6336-T3[2006] (17)		
ID=23	MID=31	$h_{Fe} = f(\gamma_e, d_{en}, m_n, z_n, u, \rho_{fPv}, d_{Fen}),$ Außenverzahnung
$h_{Fe} = \frac{m_n}{2} \cdot \left[ (\cos(\gamma_e) - \sin(\gamma_e) \cdot \tan(\alpha_{Fen})) \cdot \frac{d_{en}}{m_n} - z_n \cdot \cos\left(\frac{\pi}{3} - \theta\right) - \left( \frac{G}{\cos(\theta)} - \frac{\rho_{fPv}}{m_n} \right) \right]$		
Quelle: ISO 6336-T3[2006] (18/19)		
ID=24	MID=31	$h_{Fe} = f(\gamma_e, d_{en}, m_n, z_n, u, \rho_{fPv}, d_{Fen}),$ Innenverzahnung
$h_{Fe} = \frac{m_n}{2} \cdot \left[ (\cos(\gamma_e) - \sin(\gamma_e) \cdot \tan(\alpha_{Fen})) \cdot \frac{d_{en}}{m_n} - z_n \cdot \cos\left(\frac{\pi}{6} - \theta\right) - \sqrt{3} \cdot \left( \frac{G}{\cos(\theta)} - \frac{\rho_{fPv}}{m_n} \right) \right]$		
Quelle: ISO 6336-T3[2006] (18/19)		
ID=25	MID=32	$s_R = f(d_f, d_i)$
$s_R = \frac{d_f - d_i}{2}$		
Quelle: -		

ID=26	MID=33	$M=f(P,n)$
$M = \frac{P}{2 \cdot \pi \cdot n}$		
Quelle: -		

ID=27	MID=34	$N_L=f(n,L_h)$
$N_L = n \cdot L_h$		
Quelle: -		

ID=28	MID=35	$f_{pb}=f(f_{pt},\alpha_t)$
$f_{pb} = f_{pt} \cdot \cos(\alpha_t)$		
Quelle: DIN 3990-T1 [1987] (Abschnitt 2.3)		

ID=29	MID=36	$y_p=f(f_{pb},v,WS)$
$y_p = \begin{cases} \left( \frac{160 \cdot f_{pb}}{\frac{\sigma_{Hlim} \cdot mm^2}{N}} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ \left( \frac{160 \cdot \min(f_{pb}, 80\mu m)}{\frac{\sigma_{Hlim} \cdot mm^2}{N}} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ \left( \frac{160 \cdot \min(f_{pb}, 40\mu m)}{\frac{\sigma_{Hlim} \cdot mm^2}{N}} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.275 \cdot f_{pb}) & \text{if } (1 \leq WS \leq 2) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{pb}, 80\mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{pb}, 40\mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.075 \cdot \min(f_{pb}, 40\mu m)) & \text{if } (WS = 12 \vee 4 \leq WS \leq 5) \end{cases}$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=77	MID=71	$\sigma_{H2}=f(Z_D,\sigma_{H0},K_A,K_V,K_{H\beta},K_{H\alpha})$
$\sigma_{H2} = Z_D \cdot \sigma_{H0} \cdot \sqrt{K_A \cdot K_V \cdot K_{H\beta} \cdot K_{H\alpha}}$		
Quelle: ISO 6336-T2[2006] (4)		

ID=30	MID=37	$y_f = f(f_{\alpha}, v, WS)$
$y_f = \begin{cases} \left( \frac{160 \cdot f_{\alpha}}{\sigma_{Hlim} \cdot mm^2} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ \left( \frac{160 \cdot \min(f_{\alpha}, 80 \mu m)}{\sigma_{Hlim} \cdot mm^2} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ \left( \frac{160 \cdot \min(f_{\alpha}, 40 \mu m)}{\sigma_{Hlim} \cdot mm^2} \right) & \text{if } (WS = 3 \vee 6 \leq WS \leq 11) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.275 \cdot f_{\alpha}) & \text{if } (1 \leq WS \leq 2) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{\alpha}, 80 \mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{\alpha}, 40 \mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.075 \cdot \min(f_{\alpha}, 40 \mu m)) & \text{if } (WS = 12 \vee 4 \leq WS \leq 5) \end{cases}$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=31	MID=38	$BS = f(b_s, b)$
$BS = \begin{cases} 0.2 & \text{if } \frac{b_s}{b} < 0.2 \\ 1.2 & \text{if } \frac{b_s}{b} > 1.2 \\ \frac{b_s}{b} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (83)		

ID=32	MID=39	$SR = f(s_R, m_n)$
$SR = \begin{cases} 1 & \text{if } \frac{s_R}{m_n} < 1 \\ \frac{s_R}{m_n} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (83)		

ID=33	MID=40	$C_B=f(h_{fp}, m_n, a_p)$
$C_B = \left[ 1 + 0.5 \cdot \left( 1.25 - \frac{h_{fp}}{m_n} \right) \right] \cdot \left[ 1 - 0.02 \cdot \left( 20 - \frac{a_p}{\sigma} \right) \right]$		
Quelle: ISO 6336-T1 [2006] (86)		

ID=34	MID=41	$\eta_E=f(Z, C_{ya}, m_{red})$
$\eta_E = \frac{30000}{\pi \cdot Z} \cdot \sqrt{\frac{C_{ya} \alpha}{3600000000 m_{red}}}$		
Quelle: ISO 6336-T1 [2006] (6)		

ID=35	MID=42	$C_{ay}=f(\sigma_{Hlim})$
$C_{ay} = \frac{1}{18} \cdot \left( \frac{\sigma_{Hlim} \cdot \frac{mm^2}{N}}{97} - 18.45 \right)^2 + 1.5$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=36	MID=43	$\gamma_\beta=f(F_{\beta x}, WS, v, \sigma_{Hlim})$
$\gamma_\beta = \begin{cases} \left( \frac{320}{\sigma_{Hlim} \cdot \frac{mm^2}{N}} \cdot F_{\beta x} \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( v \leq 5 \frac{m}{s} \right) \\ \left( \frac{320}{\sigma_{Hlim} \cdot \frac{mm^2}{N}} \cdot \min(F_{\beta x}, 80 \mu m) \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ \left( \frac{320}{\sigma_{Hlim} \cdot \frac{mm^2}{N}} \cdot \min(F_{\beta x}, 40 \mu m) \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.55 \cdot F_{\beta x}) & \text{if } (1 \leq WS \leq 2) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ (0.55 \cdot \min(F_{\beta x}, 80 \mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ (0.55 \cdot \min(F_{\beta x}, 80 \mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.15 \cdot \min(F_{\beta x}, 40 \mu m)) & \text{if } [WS = 12 \vee (4 \leq WS \leq 5)] \end{cases}$		
Quelle: ISO 6336-T1 [2006] (44-50)		

ID=37	MID=44	$y_a = f(\sigma_{Hlim}, f_{pb}, WS, v)$
$y_\alpha = \begin{cases} \left( \frac{160}{\frac{\sigma_{Hlim}}{N} \cdot f_{pb}} \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( v \leq 5 \frac{m}{s} \right) \\ \left( \frac{160}{\frac{\sigma_{Hlim}}{N} \cdot \min(f_{pb}, 80\mu m)} \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ \left( \frac{160}{\frac{\sigma_{Hlim}}{N} \cdot \min(f_{pb}, 40\mu m)} \right) & \text{if } [WS = 3 \vee (6 \leq WS \leq 11)] \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.275 \cdot f_{pb}) & \text{if } (1 \leq WS \leq 2) \wedge \left( v \leq 5 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{pb}, 80\mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( 5 \frac{m}{s} < v \leq 10 \frac{m}{s} \right) \\ (0.275 \cdot \min(f_{pb}, 80\mu m)) & \text{if } (1 \leq WS \leq 2) \wedge \left( v > 10 \frac{m}{s} \right) \\ (0.075 \cdot \min(f_{pb}, 40\mu m)) & \text{if } [WS = 12 \vee (4 \leq WS \leq 5)] \end{cases}$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=38	MID=45	$Y_F = f(h_{Fe}, a_{Fen}, s_{Fn}, m_n, a_n)$
$Y_F = \frac{\frac{6 \cdot h_{Fe}}{m_n} \cdot \cos(\alpha_{Fen})}{\left( \frac{s_{Fn}}{m_n} \right)^2 \cdot \cos(\alpha_n)}$		
Quelle: ISO 6336-T3[2006] (9)		

ID=39	MID=46	$L = f(s_{Fn}, h_{Fe})$
$L = \frac{s_{Fn}}{h_{Fe}}$		
Quelle: ISO 6336-T3[2006] (37)		

ID=40	MID=47	$q_S = f(s_{Fn}, \rho_F)$
$q_S = \frac{s_{Fn}}{2 \cdot \rho_F}$		
Quelle: ISO 6336-T3[2006] (38)		



ID=41	MID=48	$Y_S=f(L,q_S)$
$Y_S = (1.2 + 0.13 \cdot L) \cdot q_S^{\left( \frac{1}{1.21 + \frac{2.3}{L}} \right)}$		
Quelle: ISO 6336-T3[2006] (36)		

ID=42	MID=49	$Y_B=f()$ , Vollrad, Außenverzahnung
$Y_B = 1$		
Quelle:		

ID=43	MID=49	$Y_B=f(s_R,h)$ , Zahnkranz, Außenverzahnung
$Y_B = \begin{cases} 1 & \text{if } \frac{s_R}{h} \geq 1.2 \\ \left( 1.6 \cdot \ln \left( 2.242 \cdot \frac{h}{s_R} \right) \right) & \text{if } 0.5 < \frac{s_R}{h} < 1.2 \end{cases}$		
Quelle: ISO 6336-T3[2006] (41-44)		

ID=44	MID=49	$Y_B=f(s_R,h)$ , Innenverzahnung
$Y_B = \begin{cases} 1 & \text{if } \frac{s_R}{m_n} \geq 3.5 \\ \left( 1.15 \cdot \ln \left( 8.324 \cdot \frac{m_n}{s_R} \right) \right) & \text{if } 1.75 < \frac{s_R}{m_n} < 3.5 \end{cases}$		
Quelle: ISO 6336-T3[2006] (41-44)		

ID=45	MID=50	$Y_{DT}=f(\epsilon_{\alpha n}, q_V)$
$Y_{DT} = \begin{cases} 1 & \text{if } \epsilon_{\alpha n} \leq 2.05 \vee (\epsilon_{\alpha n} > 2.05 \wedge q_V > 4) \\ -0.666 \cdot \epsilon_{\alpha n} + 2.366 & \text{if } 2.05 < \epsilon_{\alpha n} \leq 2.5 \wedge q_V \leq 4 \\ 0.7 & \text{if } \epsilon_{\alpha n} > 2.5 \wedge q_V \leq 4 \end{cases}$		
Quelle: ISO 6336-T3[2006] (45-47)		



ID=46	MID=51	$x_{zeit}=f(WS)$
$x_{zeit} = \begin{cases} 10^3 & \text{if } WS = 1 \vee WS = 2 \vee WS = 4 \vee WS = 5 \vee WS = 12 \\ 10^4 & \text{if } WS = 3 \vee WS = 6 \vee WS = 7 \vee WS = 8 \vee WS = 9 \vee WS = 10 \vee WS = 11 \\ 3 \cdot 10^6 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T3[2006] (12.2.1; Figure 9)		

ID=47	MID=52	$y_{zeit}=f(WS)$
$y_{zeit} = \begin{cases} 1.6 & \text{if } WS = 1 \vee WS = 2 \vee WS = 4 \\ 1.1 & \text{if } WS = 5 \\ 2.5 & \text{if } WS = 3 \vee WS = 6 \vee WS = 7 \vee WS = 8 \vee WS = 9 \vee WS = 10 \vee WS = 11 \vee WS = 12 \\ 1 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T3[2006] (12.2.1; Figure 9)		

ID=50	MID=56	$Y_{oT}=f(Y_S, R_{pN}, WS),$ Kurzzeitfestigkeit
$Y_{\delta relT} = \begin{cases} \frac{1 + 0.93 \cdot (Y_S - 1) \cdot \sqrt[4]{\frac{200}{R_{pN} \cdot \frac{mm^2}{N}}}}{1 + 0.93 \cdot \sqrt[4]{\frac{200}{R_{pN} \cdot \frac{mm^2}{N}}}} & \text{if } 6 \leq WS \leq 9 \\ \frac{1 + 0.82 \cdot (Y_S - 1) \cdot \sqrt[4]{\frac{300}{R_{pN} \cdot \frac{mm^2}{N}}}}{1 + 0.82 \cdot \sqrt[4]{\frac{300}{R_{pN} \cdot \frac{mm^2}{N}}}} & \text{if } 10 \leq WS \leq 11 \\ 0.44 \cdot Y_S + 0.12 & \text{if } WS = 12 \\ 0.2 \cdot Y_S + 0.6 & \text{if } 4 \leq WS \leq 5 \\ 0.075 \cdot Y_S + 0.85 & \text{if } WS = 3 \\ 1 & \text{if } 1 \leq WS \leq 2 \end{cases}$		
Quelle: ISO 6336-T3[2006] (48)		

ID=51	MID=56	$Y_{\delta relT} = f(\rho_{Strich}, X_{T\_}),$ Dauerfestigkeit
$Y_{\delta relT} = \frac{1 + \sqrt{\rho_{Strich} \cdot X_{Stern}}}{1 + \sqrt{\rho_{Strich} \cdot X_{T\_Stern}}}$		
Quelle: ISO 6336-T3[2006] (48)		

ID=52	MID=56	$Y_{\delta relT} = f(X_{Zeit}, Y_{\delta zeit}, N_L),$ Zeitfestigkeit
$Y_{\delta relT} = 10^{\text{interp}(\log(x_{zeit}), \log(y_{\delta zeit}), \log(N_L))}$		
Quelle: ISO 6336-T3[2006] (48)		

ID=53	MID=54	$\rho_- = f(WS)$
$\rho_- = \begin{cases} 0.3124 & \text{if } WS = 1 \\ 0.3095 & \text{if } WS = 2 \\ 0.1005 & \text{if } 3 \leq WS \leq 5 \\ 0.0833 & \text{if } WS = 6 \\ 0.0445 & \text{if } WS = 7 \\ 0.0281 & \text{if } WS = 8 \\ 0.0194 & \text{if } WS = 9 \\ 0.0064 & \text{if } WS = 10 \\ 0.0014 & \text{if } WS = 11 \\ 0.003 & \text{if } WS = 12 \end{cases}$		
Quelle: ISO 6336-T3[2006] (Table 2)		

ID=54	MID=55	$\chi_- = f(q_S)$
$\chi_- = \frac{1}{5} \cdot (1 + 2 \cdot q_S)$		
Quelle: ISO 6336-T3[2006] (50-54)		

ID=56	MID=57	$Y_{RrelT} = f(X_{Zeit}, Y_{Rzeit}, N_L),$ Zeitfestigkeit
$Y_{RrelT} = 10^{\text{interp}(\log(x_{zeit}), \log(y_{Rzeit}), \log(N_L))}$		
Quelle: ISO 6336-T3[2006] (55-60)		

ID=55	MID=57	$Y_{RelT}=f(R_{zf}, WS)$ , Dauerfestigkeit
$Y_{RelT} = \begin{cases} 1.12 & \text{if } R_{zf} < 1\mu\text{m} \wedge (WS = 3 \vee 8 \leq WS \leq 12) \\ 1.07 & \text{if } R_{zf} < 1\mu\text{m} \wedge (6 \leq WS \leq 7) \\ 1.025 & \text{if } R_{zf} < 1\mu\text{m} \wedge (1 \leq WS \leq 2 \vee 4 \leq WS \leq 5) \\ \left[ 1.674 - 0.529 \cdot \left( \frac{R_{zf}}{\mu\text{m}} + 1 \right)^{0.1} \right] & \text{if } R_{zf} \geq 1\mu\text{m} \wedge (WS = 3 \vee 8 \leq WS \leq 12) \\ \left[ 5.306 - 4.203 \cdot \left( \frac{R_{zf}}{\mu\text{m}} + 1 \right)^{0.01} \right] & \text{if } R_{zf} \geq 1\mu\text{m} \wedge (6 \leq WS \leq 7) \\ \left[ 4.299 - 3.259 \cdot \left( \frac{R_{zf}}{\mu\text{m}} + 1 \right)^{0.0058} \right] & \text{if } R_{zf} \geq 1\mu\text{m} \wedge (1 \leq WS \leq 2 \vee 4 \leq WS \leq 5) \end{cases}$		
Quelle: ISO 6336-T3[2006] (55-60)		

ID=57	MID=57	$Y_{RelT}=f()$ , Kurzzeitfestigkeit
$Y_{RelT} = 1$		
Quelle: ISO 6336-T3[2006] (55-60)		

ID=58	MID=58	$Y_X=f(m_n, WS)$ , Dauerfestigkeit
$Y_X = \begin{cases} 1 & \text{if } m_n \leq 5\text{mm} \\ \left( 1.03 - 0.006 \cdot \frac{m_n}{\text{mm}} \right) & \text{if } 5\text{mm} < m_n < 30\text{mm} \wedge (WS = 3 \vee 6 \leq WS \leq 12) \\ 0.85 & \text{if } m_n \geq 30\text{mm} \wedge (WS = 3 \vee 6 \leq WS \leq 12) \\ \left( 1.05 - 0.01 \cdot \frac{m_n}{\text{mm}} \right) & \text{if } 5\text{mm} < m_n < 25\text{mm} \wedge (WS = 12 \vee 4 \leq WS \leq 5) \\ 0.8 & \text{if } m_n > 25\text{mm} \wedge (WS = 12 \vee 4 \leq WS \leq 5) \\ \left( 1.075 - 0.015 \cdot \frac{m_n}{\text{mm}} \right) & \text{if } 5\text{mm} < m_n < 25\text{mm} \wedge (1 \leq WS \leq 2) \\ 0.7 & \text{if } m_n \geq 25\text{mm} \wedge (1 \leq WS \leq 2) \end{cases}$		
Quelle: ISO 6336-T3[2006] (Abschnitt 15.2.2)		

ID=59	MID=58	$Y_X=f(x_{zeit}, y_{Xzeit}, N_L)$ , Zeitfestigkeit
$Y_X = 10^{\text{interp}(\log(x_{zeit}), \log(y_{Xzeit}), \log(N_L))}$		
Quelle:		

ID=60	MID=59	$\sigma_{F0}=f(F_t, b, m_n, Y_F, Y_S, Y_\beta, Y_B, Y_{DT})$
$\sigma_{F0} = \frac{F_t}{b \cdot m_n} \cdot Y_F \cdot Y_S \cdot Y_\beta \cdot Y_B \cdot Y_{DT}$		
Quelle: ISO 6336-T3[2006] (4)		
ID=61	MID=60	$\sigma_{FP}(\sigma_{Flim}, Y_{ST}, Y_{NT}, S_{Fmin}, Y_{\delta relT}, Y_{RrelT}, Y_X)$
$\sigma_{FP} = \frac{\sigma_{Flim} \cdot Y_{ST} \cdot Y_{NT}}{S_{Fmin}} \cdot Y_{\delta relT} \cdot Y_{RrelT} \cdot Y_X$		
Quelle: ISO 6336-T3[2006] (5)		
ID=62	MID=61	$\sigma_{FG}=f(\sigma_{FP}, S_{Fmin})$
$\sigma_{FG} = \sigma_{FP} \cdot S_{Fmin}$		
Quelle: ISO 6336-T3[2006] (5)		
ID=63	MID=62	$\sigma_F=f(\sigma_{F0}, K_A, K_V, K_{F\beta}, K_{F\alpha})$
$\sigma_F = \sigma_{F0} \cdot K_A \cdot K_V \cdot K_{F\beta} \cdot K_{F\alpha}$		
Quelle: ISO 6336-T3[2006] (3)		
ID=64	MID=63	$S_F=f(\sigma_{FG}, \sigma_F)$
$S_F = \frac{\sigma_{FG}}{\sigma_F}$		
Quelle: ISO 6336-T3[2006] (1,2)		
ID=70	MID=65	$\rho=f(d_b, a_{wt})$
$\rho = -(0.5 \cdot d_b \cdot \tan(\alpha_{wt}))$		
Quelle: ISO 6336-T2[2006] (47)		
ID=73	MID=67	$\sigma_{HP}=f(\sigma_{Hlim}, Z_{NT}, S_{Hmin}, Z_L, Z_V, Z_R, Z_W, Z_X)$
$\sigma_{HP} = \frac{\sigma_{Hlim} \cdot Z_{NT}}{S_{Hmin}} \cdot Z_L \cdot Z_V \cdot Z_R \cdot Z_W \cdot Z_X$		
Quelle:		

ID=74	MID=68	$\sigma_{HG}=f(\sigma_{HP}, S_{Hmin})$
$\sigma_{HG} = \sigma_{HP} \cdot S_{Hmin}$		
Quelle: ISO 6336-T2[2006] (6)		
ID=75	MID=69	$S_H=f(\sigma_{HG}, \sigma_H)$
$S_H = \frac{\sigma_{HG}}{\sigma_H}$		
Quelle: ISO 6336-T2[2006] (1)		
ID=76	MID=70	$\sigma_{H1}=f(Z_B, \sigma_{H0}, K_A, K_V, K_{H\beta}, K_{H\alpha})$
$\sigma_{H1} = Z_B \cdot \sigma_{H0} \cdot \sqrt{K_A \cdot K_V \cdot K_{H\beta} \cdot K_{H\alpha}}$		
Quelle: ISO 6336-T2[2006] (4)		
ID=78	MID=78	$m_t=f(m_n, \beta)$
$m_t = \frac{m_n}{\cos(\beta)}$		
Quelle: DIN-ISO 21771[2014] (2)		
ID=79	MID=79	$\beta_b=f(\beta, \alpha_n)$
$\beta_b = \arcsin(\sin(\beta) \cdot \cos(\alpha_n))$		
Quelle: DIN-ISO 21771[2014] (6)		
ID=80	MID=80	$\alpha_t=f(\alpha_n, \beta)$
$\alpha_t = \arctan\left(\frac{\tan(\alpha_n)}{\cos(\beta)}\right)$		
Quelle: DIN-ISO 21771[2014] (14)		
ID=81	MID=81	$p_{bt}=f(d_{b1}, z_1)$
$p_{bt} = \frac{\pi \cdot d_{b1}}{ z_1 }$		
Quelle: DIN-ISO 21771[2014] (28)		

ID=82	MID=82	$p_{bn}=f(p_{bt},\beta_b)$
$p_{bn} = p_{bt} \cdot \cos(\beta_b)$		
Quelle: DIN-ISO 21771[2014] (29)		
ID=83	MID=83	$p_{et}=f(p_{bt})$
$p_{et} = p_{bt}$		
Quelle: DIN-ISO 21771[2014] (30)		
ID=84	MID=84	$p_{en}=f(p_{bn})$
$p_{en} = p_{bn}$		
Quelle: DIN-ISO 21771[2014] (31)		
ID=85	MID=85	$F_t=f(M_1,d_1)$
$F_t = \frac{2 \cdot M_1}{d_1}$		
Quelle: -		
ID=86	MID=86	$v=f(d_1,n_1)$
$v = \pi \cdot d_1 \cdot n_1$		
Quelle: -		
ID=87	MID=57	$Y_{RelT}=f()$ , Kurzzeitfestigkeit
$Y_{RelT} = 1$		
Quelle: ISO 6336-T3[2006] (61)		
ID=88	MID=58	$Y_X=f()$ , Kurzzeitfestigkeit
$Y_X = 1$		
Quelle: ISO 6336-T3[2006] (Table 3)		
ID=89	MID=87	$Z_X=f()$
$Z_X = 1$		
Quelle: ISO 6336-T2[2006] (Abschnitt 14)		

Klasse Stirnradsatz, eCl@ss 23-17-17-10

Liste der Merkmale

ID	Name	Symbol	Einheit	Funktion	Gruppe
1.	Hilfsfaktor	$K_-$	-	-	3
2.	zu übertragende Leistung	$P$	W	-	2
3.	Anwendungsfaktor	$K_A$	-	-	3
4.	Profilverschiebungssumme	$x_s$	-	1	1
5.	Stirnbetriebseingriffswinkel	$\alpha_{wt}$	°	2	1
6.	Achsabstand	$a$	mm	3	1
7.	Gemeinsame Zahnhöhe	$h_w$	mm	4	1
8.	Zähnezahlverhältnis	$u$	-	5	2
9.	Übersetzungsverhältnis	$i$	-	6	2
10.	Länge Eintritts-Eingriffsstrecke	$g_f$	mm	7,8,9	1
11.	Länge Austritts-Eingriffsstrecke	$g_a$	mm	10,11,12	1
12.	Länge Eingriffsstrecke	$g_a$	mm	13,14	1
13.	Profilüberdeckung	$\varepsilon_a$	-	15	1
14.	Sprungüberdeckung	$\varepsilon_\beta$	-	16	1
15.	Gesamtüberdeckung	$\varepsilon_\gamma$	mm	17	1
16.	Herstellungs- Flankenlinienabweichung	$f_{ma}$	$\mu m$	18	3
17.	Einlaufbetrag	$y_p$	$\mu m$	19	3
18.	Einlaufbetrag	$y_f$	$\mu m$	20	3
19.	Effektive Grundkreisabweichung	$f_{pb\_eff}$	$\mu m$	21	3
20.	Effektive Profilformabweichung	$f_{fa\_eff}$	$\mu m$	22	3
21.	Theoretische Einzelfedersteifigkeit	$c_{th\_}$	N/(mm* $\mu m$ )	23,24	3
22.	Korrekturfaktor	$C_M$	-	25,26	3
23.	Radkörperfaktor	$C_R$	-	27,28	3
24.	Bezugsprofilfaktor	$C_B$	-	29	3
25.	Einzelfedersteifigkeit	$c_-$	N/(mm* $\mu m$ )	30	3
26.	Eingriffsfedersteifigkeit Geradstirnräder	$c_{\gamma a}$	N/(mm* $\mu m$ )	31	3
27.	Eingriffsfedersteifigkeit Schrägstirnräder	$c_{\gamma \beta}$	N/(mm* $\mu m$ )	32	3
28.	Auf Ritzel reduzierte Masse	$m_{red}$	kg/mm	33,34	3
29.	Resonanzverhältnis	$N_B$	-	35	3
30.	Untergrenze Resonanzverhältnis	$N_S$	-	36	3

31.	Kontante Dynamikfaktor $C_{v1}$	$C_{v1}$	-	37	3
32.	Kontante Dynamikfaktor $C_{v2}$	$C_{v2}$	-	38	3
33.	Kontante Dynamikfaktor $C_{v3}$	$C_{v3}$	-	39	3
34.	Kontante Dynamikfaktor $C_{v4}$	$C_{v4}$	-	40	3
35.	Kontante Dynamikfaktor $C_{v5}$	$C_{v5}$	-	41	3
36.	Kontante Dynamikfaktor $C_{v6}$	$C_{v6}$	-	42	3
37.	Kontante Dynamikfaktor $C_{v7}$	$C_{v7}$	-	43	3
38.	Kontante Dynamikfaktor $C_{ay}$	$C_{ay}$	-	44	3
39.	Auslegungswert Profilkorrektur	$C_{a\_nenn}$	-	45	3
40.	Kontante Dynamikfaktor $C_a$	$C_a$	-	46	3
41.	Kontante Dynamikfaktor $B_p$	$B_p$	-	47	3
42.	Kontante Dynamikfaktor $B_f$	$B_f$	-	48	3
43.	Kontante Dynamikfaktor $B_k$	$B_k$	-	49	3
44.	Faktor für Lastaufteilung	$B_-$	-	50	3
45.	Flankenlinienabweichung durch Ritzelverformung	$f_{sh}$	-	51	3
46.	gesamte Flankenlinienabweichung vor Einlauf	$F_{\beta x}$	$\mu m$	52	3
47.	Einlaufbetrag Flankenlinienwinkelabweichung	$\gamma_\beta$	$\mu m$	53	3
48.	Gesamte Flankenlinienwinkelabweichung nach Einlauf	$F_{\beta y}$	$\mu m$	54	3
49.	Breitenfaktor (Flanke)	$K_{H\beta}$	-	55	3
50.	Breitenfaktor (Fuß)	$K_{F\beta}$	-	56	3
51.	Einlaufbetrag Flankenlinienabweichung	$\gamma_\alpha$	$\mu m$	57	3
52.	Überdeckungsfaktor	$Z_\epsilon$	-	58	3
53.	Stirnfaktor (Flanke)	$K_{Ha}$	-	59	3
54.	Stirnfaktor (Fuß)	$K_{Fa}$	-	60	3
55.	Referenzschrägungswinkel	$\beta_{ref}$	°	61	4
56.	Schrägenfaktor	$Y_\beta$	-	62	4
57.	Referenzspannungsfaktor	$Y_{ST}$	-	63	4
58.	Bezogenes Spannungsgefälle	$\chi_{T-}$	-	64	4
59.	Zonenfaktor	$Z_H$	-	65	5
60.	Elastizitätsfaktor	$Z_E$	-	66	5
61.	Schrägenfaktor	$Z_\beta$	-	67	5
62.	Hilfsfaktor $M_{11}$	$M_{11}$	-	68	5



63.	Hilfsfaktor $M_{22}$	$M_{22}$	-	69	5
64.	Eingriffsfaktor	$Z_B$	-	70	5
65.	Eingriffsfaktor	$Z_D$	-	71	5
66.	Hilfswert Schmierstofffaktor	$C_{ZL}$	-	72	5
67.	Schmierstofffaktor	$Z_L$	-	73	5
68.	Geschwindigkeitsfaktor	$Z_V$	-	74	5
69.	Mittlere Rauhtiefe (Flanke)	$R_z$	$\mu\text{m}$	75	5
70.	Hilfswert für Rauheitsfaktor (Flanke)	$C_{ZR}$	-	76	5
71.	Rauheitsfaktor	$Z_R$	-	77	5
72.	Äquivalente Rauigkeit	$R_{zH}$	$\mu\text{m}$	78	5
73.	Zahnflankenpressung	$\sigma_{H0}$	$\text{N/mm}^2$	79	5
74.	Mittlere relative Rauhtiefe	$R_{z10}$	$\mu\text{m}$	80	5
75.	Dynamikfaktor	$K_V$	-	81	3
76.	Referenzsprungüberdeckung	$\varepsilon_{\beta\_ref}$	°	82	4

## Liste der Funktionsbausteine

ID=1	MID=4	$x_s = f(x_1, x_2)$
$x_s = x_1 + x_2$		
Quelle: DIN-ISO 21771[2014] (2)		
ID=2	MID=5	$\alpha_{wt} = f(\alpha_t, \alpha_n, z_1, z_2, x_1, x_2)$
$\tan(\alpha_{wt}) - \alpha_{wt} = \tan(\alpha_t) - \alpha_t + \frac{2 \cdot \tan(\alpha_n)}{z_1 + z_2} \cdot (x_1 + x_2)$		
Quelle: DIN-ISO 21771[2014] (55)		
ID=3	MID=6	$a = f(z_1, z_2, \alpha_t, \alpha_{wt}, \beta)$
$a = \frac{ z_1 + z_2  \cdot m_n \cdot \cos(\alpha_t)}{2 \cdot \cos(\alpha_{wt}) \cdot \cos(\beta)}$		
Quelle:		
ID=4	MID=7	$h_w = f(z_1, z_2, d_{a1}, d_{a2}, a_w)$
$h_w = \frac{\frac{z_1}{ z_1 } \cdot d_{a1} + \frac{z_2}{ z_2 } \cdot d_{a2}}{2} - \frac{z_1}{ z_1 } \cdot \frac{z_2}{ z_2 } \cdot a$		
Quelle: DIN-ISO 21771[2014] (54)		

ID=5	MID=8	$u=f(z_1, z_2)$
$u = \frac{z_2}{z_1}$		
Quelle: -		

ID=6	MID=9	$i=f(u, z_1, z_2)$
$i = \begin{cases} u & \text{if } z_2 > z_1 \\ \frac{1}{u} & \text{if } z_2 \leq z_1 \end{cases}$		
Quelle: -		

ID=7	MID=10	$g_f=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Außenverzahnung, Antrieb Rad 1
$g_f = \frac{1}{2} \cdot \left( \sqrt{d_{Na2}^2 - d_{b2}^2} - d_{b2} \cdot \tan(\alpha_{wt}) \right)$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=8	MID=10	$g_f=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Innenverzahnung Rad 2, Antrieb Rad 1
$g_f = -\frac{1}{2} \cdot \left( \sqrt{d_{Na2}^2 - d_{b2}^2} - d_{b2} \cdot \tan(\alpha_{wt}) \right)$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=9	MID=10	$g_f=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Antrieb Rad 2
$g_f = \frac{1}{2} \cdot \left( \sqrt{d_{Na1}^2 - d_{b1}^2} - d_{b1} \cdot \tan(\alpha_{wt}) \right)$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=10	MID=11	$g_a=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Antrieb Rad 1
$g_a = \left[ \frac{1}{2} \cdot \left( \sqrt{d_{Na1}^2 - d_{b1}^2} - d_{b1} \cdot \tan(\alpha_{wt}) \right) \right]$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=11	MID=11	$g_a=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Antrieb Rad 2, Außenverzahnung Rad 2
$g_a = \left[ \frac{1}{2} \cdot \left( \sqrt{d_{Na2}^2 - d_{b2}^2} - d_{b2} \cdot \tan(\alpha_{wt}) \right) \right]$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=12	MID=11	$g_a=f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, \alpha_{wt}),$ Antrieb Rad 2, Innenverzahnung Rad 2
$g_a = \left[ -\frac{1}{2} \cdot \left( \sqrt{d_{Na2}^2 - d_{b2}^2} - d_{b2} \cdot \tan(\alpha_{wt}) \right) \right]$		
Quelle: DIN-ISO 21771[2014] (79/80)		

ID=13	MID=12	$g_a = f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, a_{wt}, a),$ Außenverzahnung
$g_\alpha = \left[ \frac{1}{2} \cdot \left( \sqrt{d_{Na1}^2 - d_{b1}^2} + \sqrt{d_{Na2}^2 - d_{b2}^2} - 2 \cdot a \cdot \sin(\alpha_{wt}) \right) \right]$		
Quelle: DIN-ISO 21771[2014] (77)		

ID=14	MID=12	$g_a = f(d_{Na2}, d_{b2}, d_{Na1}, d_{b1}, a_{wt}, a),$ Innenverzahnung Rad 2
$g_\alpha = \left[ \frac{1}{2} \cdot \left( \sqrt{d_{Na1}^2 - d_{b1}^2} - \sqrt{d_{Na2}^2 - d_{b2}^2} + 2 \cdot a \cdot \sin(\alpha_{wt}) \right) \right]$		
Quelle: DIN-ISO 21771[2014] (77)		

ID=15	MID=13	$\epsilon_a = f(g_f, g_a, p_{et})$
$\epsilon_\alpha = \frac{g_f + g_a}{p_{et}}$		
Quelle: DIN-ISO 21771[2014] (90)		

ID=16	MID=14	$\epsilon_\beta = f(b_1, b_2, \beta_b, p_{et})$
$\epsilon_\beta = \frac{\min(b_1, b_2) \cdot \tan(\beta_b)}{p_{et}}$		
Quelle: DIN-ISO 21771[2014] (93)		

ID=17	MID=15	$\epsilon_\gamma = f(\epsilon_a, \epsilon_\beta)$
$\epsilon_\gamma = \epsilon_\alpha + \epsilon_\beta$		
Quelle: DIN-ISO 21771[2014] (97)		

ID=18	MID=16	$f_{ma}=f(f_{H\beta 1}, f_{H\beta 2})$
$f_{ma} = \sqrt{f_{H\beta 1}^2 + f_{H\beta 2}^2}$		
Quelle:		

ID=19	MID=17	$y_p=f(y_{p1}, y_{p2})$
$y_p = 0.5 \cdot (y_{p1} + y_{p2})$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=20	MID=18	$y_f=f(y_{f1}, y_{f2})$
$y_f = 0.5 \cdot (y_{f1} + y_{f2})$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=21	MID=19	$f_{pb\_eff}=f(f_{pb1}, f_{pb2}, y_p)$
$f_{pb\_eff} = \max(f_{pb1}, f_{pb2}) - y_p$		
Quelle: ISO 6336-T1 [2006] (18)		

ID=22	MID=20	$f_{fa\_eff}=f(f_{fa1}, f_{fa2}, y_f)$
$f_{fa\_eff} = \max(f_{fa1}, f_{fa2}) - y_f$		
Quelle: ISO 6336-T1 [2006] (19)		

ID=24	MID=21	$c_{th\_} = f(z_{n1}, z_{n2}, x_1, x_2)$ , Rad 2 Innenverzahnung
$c_{th\_} = \frac{N}{\text{mm} \cdot \mu\text{m}} \cdot \frac{1}{0.04723 + \frac{0.15551}{z_{n1}} - 0.00635 \cdot x_1 - \frac{0.11654 \cdot x_1}{z_{n1}} - 0.00193 \cdot x_2 + 0.00529 \cdot x_1^2 + 0.00182 \cdot x_2^2}$		
Quelle: ISO 6336-T1 [2006] (82 / Table 9 / 9.3.1.5 b)		

ID=25	MID=22	$C_M = f()$ , Vollräder
$C_M = 0.8$		
Quelle: ISO 6336-T1 [2006] (83)		

ID=26	MID=22	$C_M = f()$ , Rad 1 oder Rad 2 Zahnkranz
$C_M = 1$		
Quelle: ISO 6336-T1 [2006] (83)		

ID=27	MID=23	$C_R = f()$ , Vollräder
$C_R = 1$		
Quelle: ISO 6336-T1 [2006] (84 / 85)		

ID=28	MID=23	$C_R = f(BS_2, SR_2)$ , Rad 1 oder Rad 2 Zahnkranz
$C_R = 1 + \frac{\ln(BS_2)}{\frac{SR_2}{5e^5}}$		
Quelle: ISO 6336-T1 [2006] (84 / 85)		

ID=29	MID=24	$C_B = f(C_{B1}, C_{B2})$
$C_B = 0.5 \cdot (C_{B1} + C_{B2})$		
Quelle: ISO 6336-T1 [2006] (87)		

ID=31	MID=26	$C_{\gamma\alpha} = f(c_-, \epsilon_\alpha, \beta)$
$c_{\gamma\alpha} = \begin{cases} [c_- \cdot (0.75 \cdot \epsilon_\alpha + 0.25)] & \text{if } (\epsilon_\alpha \geq 1.2) \wedge \beta \leq 30^\circ \\ [c_- \cdot (0.75 \cdot \epsilon_\alpha + 0.25) \cdot 0.9] & \text{if } (\epsilon_\alpha < 1.2) \end{cases}$		
Quelle: ISO 6336-T1 [2006] (91)		

ID=32	MID=27	$C_{\gamma\beta} = f(C_{\gamma\alpha})$
$C_{\gamma\beta} = 0.85 \cdot C_{\gamma\alpha}$		
Quelle:		

ID=33	MID=28	$m_{red} = f(d_{a1}, d_{f1}, d_{b1}, \rho_1, \rho_2, d_{i1}, d_{a2}, d_{f2}, d_{b2}, d_{i2}), \text{ Rad 1 und Rad 2 Vollräder}$
$m_{red} = \frac{\pi}{8} \cdot \left( \frac{d_{a1} + d_{f1}}{2 \cdot d_{b1}} \right)^2 \cdot \frac{[0.5 \cdot (d_{a1} + d_{f1})]^2}{\frac{1}{\rho_1} + \frac{1}{\rho_2 \cdot u^2}}$		
Quelle: ISO 6336-T1 [2006] (32)		



ID=34	MID=28	$m_{\text{red}} = f(d_{a1}, d_{f1}, d_{b1}, \rho_1, \rho_2, d_{i1}, d_{a2}, d_{f2}, d_{b2}, d_{i2}), \text{ Rad 1 und Rad 2 Radkranz}$
$m_{\text{red}} = \frac{\pi}{8} \cdot \left( \frac{d_{a1} + d_{f1}}{2 \cdot d_{b1}} \right)^2 \cdot \frac{[0.5 \cdot (d_{a1} + d_{f1})]^2}{\frac{1}{\left[ 1 - \left[ \frac{d_{i1}}{0.5 \cdot (d_{a1} + d_{f1})} \right]^4 \right] \cdot \rho_1} + \frac{1}{\left[ 1 - \left[ \frac{d_{i2}}{0.5 \cdot (d_{a2} + d_{f2})} \right]^4 \right] \cdot \rho_2 \cdot u^2}}$		
Quelle: ISO 6336-T1 [2006] (32)		

ID=35	MID=29	$N_B = f(n_1, n_{E1})$
$N_B = \frac{n_1}{n_{E1}}$		
Quelle: ISO 6336-T1 [2006] (9)		

ID=36	MID=30	$N_S = f(F_t, N, K_A, b_1, b_2)$
$N_S = \begin{cases} \left( 0.5 + 0.35 \cdot \sqrt{\frac{\frac{F_t}{N} \cdot K_A}{b_1}} \right) & \text{if } \left( \frac{F_t \cdot K_A}{\min(b_1, b_2)} \right) < 100 \frac{\text{N}}{\text{mm}} \\ 0.85 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (11/12)		

ID=37	MID=31	$C_{v1} = f()$
$C_{v1} = 0.32$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=38	MID=32	$C_{V2}=f(\epsilon_V)$
$C_{V2} = \begin{cases} 0.34 & \text{if } 1 < \epsilon_V \leq 2 \\ \frac{0.57}{\epsilon_V - 0.3} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=39	MID=33	$C_{V3}=f(\epsilon_V)$
$C_{V3} = \begin{cases} 0.23 & \text{if } 1 < \epsilon_V \leq 2 \\ \frac{0.096}{\epsilon_V - 1.56} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=40	MID=34	$C_{V4}=f(\epsilon_V)$
$C_{V4} = \begin{cases} 0.9 & \text{if } 1 < \epsilon_V \leq 2 \\ \frac{0.57 - 0.05 \cdot \epsilon_V}{\epsilon_V - 1.44} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=41	MID=35	$C_{V5}=f()$
$C_{V5} = 0.47$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=42	MID=36	$C_{v6}=f(\varepsilon_v)$
$C_{v6} = \begin{cases} 0.47 & \text{if } 1 < \varepsilon_v \leq 2 \\ \frac{0.12}{\varepsilon_v - 1.74} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=43	MID=37	$C_{v7}=f(\varepsilon_v)$
$C_{v7} = \begin{cases} 0.75 & \text{if } 1 < \varepsilon_v \leq 1.5 \\ 1 & \text{if } \varepsilon_v > 2.5 \\ 0.125 \cdot \sin[\pi \cdot (\varepsilon_v - 2)] + 0.875 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=44	MID=38	$C_{ay}=f(C_{ay1}, C_{ay2})$
$C_{ay} = 0.5 \cdot (C_{ay1} + C_{ay2})$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=45	MID=39	$C_{a\_nenn}=f(C_{a1\_nenn}, C_{a2\_nenn})$
$C_{a\_nenn} = 0.5 \cdot (C_{a1\_nenn} + C_{a2\_nenn})$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=46	MID=40	$C_a = (C_{ay}, C_{a1\_nenn}, C_{a2\_nenn}, C_{a\_nenn})$
$C_a = \begin{cases} C_{ay} & \text{if } (C_{a1\_nenn} = 0) \wedge (C_{a2\_nenn} = 0) \\ C_{a\_nenn} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=47	MID=41	$B_p = f(c_-, f_{pb\_eff}, K_A, b_1, b_2)$
$B_p = \frac{c_- \cdot f_{pb\_eff}}{K_A \cdot \frac{F_t}{\min(b_1, b_2)}}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=48	MID=42	$B_f = f(c_-, f_{fa\_eff}, K_A, b_1, b_2)$
$B_f = \frac{c_- \cdot f_{fa\_eff}}{K_A \cdot \frac{F_t}{\min(b_1, b_2)}}$		
Quelle: ISO 6336-T1 [2006] (Table 4)		

ID=49	MID=43	$B_k = f(c_-, C_a, K_A, b_1, b_2, q_{V1}, q_{V2})$
$B_k = \begin{cases} \left  1 - \frac{c_- \cdot C_a \cdot mm}{K_A \cdot \frac{F_t}{\min(b_1, b_2)}} \right  & \text{if } (q_{V1} < 6) \wedge (q_{V2} < 6) \\ 1 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (13-22)		

ID=50	MID=44	$B_{-}=f()$
$B_{-} = 1$		
Quelle: ISO 6336-T1 [2006] (7.5.2.4.1)		

ID=51	MID=45	$f_{sh}=f(F_t, K_A, K_V, b_1, b_2, B_{-}, K_{-}, l_L, s_1, d_1, d_{sh})$
$f_{sh} = \frac{1000 \mu m^2 \cdot F_t \cdot K_A \cdot K_V}{N \cdot \min(b_1, b_2)} \cdot 0.023 \cdot \left[ B_{-} + K_{-} \cdot \frac{l_L \cdot s_1}{d_1^2} \cdot \left( \frac{d_1}{d_{sh}} \right)^4 - 0.3 \right] + 0.3 \cdot \left( \frac{\min(b_1, b_2)}{d_1} \right)^2$		
Quelle: ISO 6336-T1 [2006] (57)		

ID=52	MID=46	$F_{\beta x}=f(f_{sh}, f_{ma})$
$F_{\beta x} = 1.33 \cdot f_{sh} + f_{ma}$		
Quelle: ISO 6336-T1 [2006] (52)		

ID=53	MID=47	$y_{\beta}=f(y_{\beta 1}, y_{\beta 2})$
$y_{\beta} = \frac{y_{\beta 1} + y_{\beta 2}}{2}$		
Quelle: ISO 6336-T1 [2006] (44-50)		

ID=54	MID=48	$F_{\beta y}=f(F_{\beta x}, y_{\beta})$
$F_{\beta y} = F_{\beta x} - y_{\beta}$		
Quelle: ISO 6336-T1 [2006] (43)		

ID=55	MID=49	$K_{H\beta} = f(F_{\beta y}, c_{\gamma\beta}, b_1, b_2, F_t, K_A, K_V)$
$K_{H\beta} = \begin{cases} \sqrt{\frac{2 \cdot F_{\beta y} \cdot c_{\gamma\beta} \cdot \min(b_1, b_2)}{F_t \cdot K_A \cdot K_V}} & \text{if } \frac{F_{\beta y} \cdot c_{\gamma\beta} \cdot \min(b_1, b_2)}{2 \cdot F_t \cdot K_A \cdot K_V} \geq 1 \\ 1 + \frac{F_{\beta y} \cdot c_{\gamma\beta} \cdot \min(b_1, b_2)}{2 \cdot F_t \cdot K_A \cdot K_V} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (39 / 41)		

ID=56	MID=50	$K_{F\beta} = f(K_{H\beta}, b_1, b_2, h_1, h_2)$
$K_{F\beta} = K_{H\beta} \frac{\left( \max \left( 3, \min \left( \frac{b_1}{h_1}, \frac{b_2}{h_2} \right) \right) \right)^2}{1 + \max \left( 3, \min \left( \frac{b_1}{h_1}, \frac{b_2}{h_2} \right) \right) + \left( \max \left( 3, \min \left( \frac{b_1}{h_1}, \frac{b_2}{h_2} \right) \right) \right)^2}$		
Quelle: ISO 6336-T1 [2006] (69 / 70)		

ID=57	MID=51	$\gamma_\alpha = f(\gamma_{\beta 1}, \gamma_{\beta 2})$
$\gamma_\alpha = \frac{\gamma_{\beta 1} + \gamma_{\beta 2}}{2}$		
Quelle: ISO 6336-T1 [2006] (75-78)		

ID=58	MID=52	$Z_\varepsilon = f(\varepsilon_\alpha, \varepsilon_\beta, \beta)$
$Z_\varepsilon = \begin{cases} \sqrt{\frac{4 - \varepsilon_\alpha}{3}} & \text{if } \beta = 0 \\ \sqrt{\frac{4 - \varepsilon_\alpha}{3} \cdot (1 - \varepsilon_\beta) + \frac{\varepsilon_\alpha}{\varepsilon_\beta}} & \text{if } \beta \neq 0 \wedge \varepsilon_\beta < 1 \\ \sqrt{\frac{1}{\varepsilon_\alpha}} & \text{if } \beta \neq 0 \wedge \varepsilon_\beta \geq 1 \end{cases}$		
Quelle: ISO 6336-T2 [2006] (24-26)		

ID=59	MID=53	$K_{H\alpha} = f(c_{y\alpha}, f_{pb1}, f_{pb2}, b_1, b_2, F_t, K_A, K_V, K_{H\beta}, \varepsilon_\gamma, \varepsilon_\alpha, \gamma_\alpha)$
$K_{H\alpha} = \begin{cases} \max \left[ 1, \begin{cases} \frac{\varepsilon_\gamma}{2} \cdot \left[ 0.9 + 0.4 \cdot \frac{c_{y\alpha} \cdot (\max(f_{pb1}, f_{pb2}) - \gamma_\alpha) \cdot \min(b_1, b_2)}{F_t \cdot K_A \cdot K_V \cdot K_{H\beta}} \right] & \text{if } \varepsilon_\gamma \leq 2 \\ \left[ 0.9 + 0.4 \cdot \sqrt{\frac{2 \cdot (\varepsilon_\gamma - 1)}{\varepsilon_\gamma}} \cdot \frac{c_{y\alpha} \cdot (\max(f_{pb1}, f_{pb2}) - \gamma_\alpha) \cdot \min(b_1, b_2)}{F_t \cdot K_A \cdot K_V \cdot K_{H\beta}} \right] & \text{if } \varepsilon_\gamma > 2 \end{cases} \right] & \text{if } K_{H\alpha\_vor} \leq \frac{\varepsilon_\gamma}{\varepsilon_\alpha \cdot Z_\varepsilon^2} \\ \frac{\varepsilon_\gamma}{\varepsilon_\alpha \cdot Z_\varepsilon^2} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T1 [2006] (71-73)		

ID=60	MID=54	$K_{Fa} = f(c_{y\alpha}, f_{pb1}, f_{pb2}, b_1, b_2, F_t, K_A, K_V, K_{H\beta}, \varepsilon_\gamma, \varepsilon_\alpha, \gamma_\alpha)$
$K_{Fa} = \begin{cases} \max \left[ 1, \begin{cases} \frac{\varepsilon_\gamma}{2} \cdot \left[ 0.9 + 0.4 \cdot \frac{c_{y\alpha} \cdot (\max(f_{pb1}, f_{pb2}) - \gamma_\alpha) \cdot \min(b_1, b_2)}{F_t \cdot K_A \cdot K_V \cdot K_{H\beta}} \right] & \text{if } \varepsilon_\gamma \leq 2 \\ \left[ 0.9 + 0.4 \cdot \sqrt{\frac{2 \cdot (\varepsilon_\gamma - 1)}{\varepsilon_\gamma}} \cdot \frac{c_{y\alpha} \cdot (\max(f_{pb1}, f_{pb2}) - \gamma_\alpha) \cdot \min(b_1, b_2)}{F_t \cdot K_A \cdot K_V \cdot K_{H\beta}} \right] & \text{if } \varepsilon_\gamma > 2 \end{cases} \right] & \text{if } K_{H\alpha\_vor} \leq \frac{\varepsilon_\gamma}{0.25 \cdot \varepsilon_\alpha + 0.75} \\ \frac{\varepsilon_\gamma}{0.25 \cdot \varepsilon_\alpha + 0.75} & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T3[2006] (8.2)		

ID=61	MID=55	$\beta_{\text{ref}}=f(\beta)$
$\beta_{\text{ref}} = \begin{cases} \beta & \text{if } \beta \leq 30^\circ \\ (30^\circ) & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T3[2006] (8.2)		

ID=62	MID=56	$Y_\beta(\epsilon_{\beta_{\text{ref}}}, \beta_{\text{ref}})$
$Y_\beta = 1 - \epsilon_{\beta_{\text{ref}}} \cdot \frac{\beta_{\text{ref}}}{120^\circ}$		
Quelle: ISO 6336-T3[2006] (40)		

ID=63	MID=57	$Y_{\text{ST}}=f()$
$Y_{\text{ST}} = 2$		
Quelle: ISO 6336-T3[2006] (7.4)		

ID=64	MID=58	$\chi_{\text{T}_-}=f()$
$\chi_{\text{T}_-} = 1.2$		
Quelle: ISO 6336-T3[2006] (49)		

ID=65	MID=59	$Z_H=f(\beta_b, \alpha_{\text{wt}}, \alpha_t)$
$Z_H = \sqrt{\frac{2 \cdot \cos(\beta_b) \cdot \cos(\alpha_{\text{wt}})}{(\cos(\alpha_t))^2 \cdot \sin(\alpha_{\text{wt}})}}$		
Quelle: ISO 6336-T2[2006] (16)		



ID=66	MID=60	$Z_E=f(u_1, E_1, u_2, E_2)$
$Z_E = \sqrt{\frac{1}{\pi \cdot \frac{N}{\text{mm}^2} \cdot \left( \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2} \right)}}$		
Quelle: ISO 6336-T2[2006] (19)		

ID=67	MID=61	$Z_\beta=(\beta)$
$Z_\beta = \frac{1}{\sqrt{\cos(\beta)}}$		
Quelle: ISO 6336-T2[2006] (36)		

ID=68	MID=62	$M_{11}=f(d_{a1}, d_{b1}, z_1, d_{a2}, d_{b2}, \epsilon_\alpha, z_2)$
$M_{11} = \frac{\tan(\alpha_{wt})}{\sqrt{\left( \sqrt{\frac{d_{a1}^2}{d_{b1}^2} - 1} - \frac{2 \cdot \pi}{z_1} \right) \cdot \left( \sqrt{\frac{d_{a2}^2}{d_{b2}^2} - 1} - (\epsilon_\alpha - 1) \cdot \frac{2 \cdot \pi}{z_2} \right)}}$		
Quelle: ISO 6336-T2[2006] (17)		

ID=69	MID=63	$M_{22}=f(d_{a1}, d_{b1}, z_1, d_{a2}, d_{b2}, \epsilon_\alpha, z_2)$
$M_{22} = \frac{\tan(\alpha_{wt})}{\sqrt{\left( \sqrt{\frac{d_{a2}^2}{d_{b2}^2} - 1} - \frac{2 \cdot \pi}{z_2} \right) \cdot \left( \sqrt{\frac{d_{a1}^2}{d_{b1}^2} - 1} - (\epsilon_\alpha - 1) \cdot \frac{2 \cdot \pi}{z_1} \right)}}$		
Quelle: ISO 6336-T2[2006] (18)		

ID=70	MID=64	$Z_B = f(M_{11}, \beta, \varepsilon_\alpha, \varepsilon_\beta)$
$Z_B = \begin{cases} \max(1, M_{11}) & \text{if } \beta = 0 \wedge \varepsilon_\alpha > 1 \\ 1 & \text{if } \beta \neq 0 \wedge (\varepsilon_\alpha > 1 \wedge \varepsilon_\beta \geq 1) \\ [M_{11} - \varepsilon_\beta \cdot (M_{11} - 1)] & \text{if } \beta \neq 0 \wedge (\varepsilon_\alpha > 1 \wedge \varepsilon_\beta < 1) \\ 1 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T2[2006] (6.2)		

ID=71	MID=65	$Z_D = f(M_{22}, \beta, \varepsilon_\alpha, \varepsilon_\beta)$
$Z_D = \begin{cases} \max(1, M_{22}) & \text{if } \beta = 0 \wedge \varepsilon_\alpha > 1 \\ 1 & \text{if } \beta \neq 0 \wedge (\varepsilon_\alpha > 1 \wedge \varepsilon_\beta \geq 1) \\ [M_{22} - \varepsilon_\beta \cdot (M_{22} - 1)] & \text{if } \beta \neq 0 \wedge (\varepsilon_\alpha > 1 \wedge \varepsilon_\beta < 1) \\ 1 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T2[2006] (6.2)		

ID=72	MID=66	$C_{ZL} = f(\sigma_{Hlim1}, \sigma_{Hlim2})$
$C_{ZL} = \begin{cases} \left( \frac{\min(\sigma_{Hlim1}, \sigma_{Hlim2}) \cdot \text{mm}^2}{4375 \cdot N} + 0.6357 \right) & \text{if } 850 \frac{N}{\text{mm}^2} \leq \min(\sigma_{Hlim1}, \sigma_{Hlim2}) \leq 1200 \frac{N}{\text{mm}^2} \\ 0.83 & \text{if } \min(\sigma_{Hlim1}, \sigma_{Hlim2}) < 850 \frac{N}{\text{mm}^2} \\ 0.91 & \text{if } \min(\sigma_{Hlim1}, \sigma_{Hlim2}) > 1200 \frac{N}{\text{mm}^2} \end{cases}$		
Quelle: ISO 6336-T2[2006] (38-40)		

ID=73	MID=67	$Z_L = f(C_{ZL}, u_{40})$
$Z_L = C_{ZL} + \frac{4 \cdot (1 - C_{ZL})}{\left(1.2 + \frac{134 \cdot \text{mm}^2}{v_{40} \cdot s}\right)^2}$		
Quelle: ISO 6336-T2[2006] (37)		

ID=74	MID=68	$Z_V = f(C_{ZL}, v, s)$
$Z_V = (C_{ZL} + 0.02) + \frac{2 \cdot [1 - (C_{ZL} + 0.02)]}{\sqrt{0.8 + \frac{32 \cdot \text{m}}{v \cdot s}}}$		
Quelle: ISO 6336-T2[2006] (42/43)		

ID=75	MID=69	$R_z = f(R_{zF1}, R_{zF2})$
$R_z = \frac{R_{zF1} + R_{zF2}}{2}$		
Quelle: ISO 6336-T2[2006] (44)		

ID=76	MID=70	$C_{ZR} = f(\sigma_{Hlim1}, \sigma_{Hlim2})$
$C_{ZR} = \begin{cases} \left(0.32 - 0.0002 \cdot \min(\sigma_{Hlim1}, \sigma_{Hlim2}) \cdot \frac{\text{mm}^2}{N}\right) & \text{if } 850 \frac{N}{\text{mm}^2} \leq \min(\sigma_{Hlim1}, \sigma_{Hlim2}) \leq 1200 \frac{N}{\text{mm}^2} \\ 0.15 & \text{if } \min(\sigma_{Hlim1}, \sigma_{Hlim2}) < 850 \frac{N}{\text{mm}^2} \\ 0.08 & \text{if } \min(\sigma_{Hlim1}, \sigma_{Hlim2}) > 1200 \frac{N}{\text{mm}^2} \end{cases}$		
Quelle: ISO 6336-T2[2006] (49-51)		

ID=77	MID=71	$Z_R = f(R_{z10})$
$Z_R = \left( \frac{3 \cdot \mu\text{m}}{R_{z10}} \right)^{C_{ZR}}$		
Quelle: ISO 6336-T2[2006] (48)		

ID=78	MID=72	$R_{zH} = f()$
$R_{zH} = \min \left[ 16 \mu\text{m}, \max \left[ 3 \mu\text{m}, \frac{R_{zFI1} \cdot \left( \frac{10 \cdot \text{mm}}{p_{\text{red}}} \right)^{0.33} \cdot \left( \frac{R_{zFI1}}{R_{zFI2}} \right)^{0.66}}{\left( \frac{v_{40} \cdot v \cdot s^2}{1.5 \text{m} \cdot \text{mm}^2} \right)^{0.33}} \right] \right]$		
Quelle: ISO 6336-T2[2006] (53)		

ID=79	MID=73	$\sigma_{H0} = f(Z_H, Z_E, Z_\varepsilon, Z_\beta, F_t, N, d_1, b_1, b_2, u)$
$\sigma_{H0} = Z_H \cdot Z_E \cdot Z_\varepsilon \cdot Z_\beta \cdot \sqrt{\frac{F_t \cdot N}{d_1 \cdot \min(b_1, b_2) \cdot \text{mm}^2} \cdot \frac{u+1}{u}}$		
Quelle: ISO 6336-T2[2006] (3)		

ID=80	MID=74	$R_{z10} = f(R_z, p_{\text{red}})$
$R_{z10} = R_z \cdot \sqrt[3]{\frac{10 \cdot \text{mm}}{p_{\text{red}}}}$		
Quelle: ISO 6336-T2[2006] (44)		

ID=81	MID=75	$K_V = f(N_B, N_S, C_{V1}, C_{V2}, C_{V3}, C_{V4}, C_{V5}, C_{V6}, C_{V7}, B_f, B_p, B_k)$
$K_V = \begin{cases} N_B \cdot [(C_{V1} \cdot B_p) + (C_{V2} \cdot B_f) + (C_{V3} \cdot B_k)] + 1 & \text{if } (N_B \leq N_S) \\ (C_{V1} \cdot B_p) + (C_{V2} \cdot B_f) + (C_{V4} \cdot B_k) + 1 & \text{if } (N_S < N_B \leq 1.15) \\ (C_{V5} \cdot B_p) + (C_{V6} \cdot B_f) + C_{V7} & \text{if } (N_B \geq 1.5) \\ \left[ (C_{V5} \cdot B_p) + (C_{V6} \cdot B_f) + C_{V7} + \frac{[(C_{V1} \cdot B_p) + (C_{V2} \cdot B_f) + (C_{V4} \cdot B_k) + 1] - [(C_{V5} \cdot B_p) + (C_{V6} \cdot B_f) + C_{V7}]}{0.35} \cdot (1.5 - N_B) \right] & \text{if } (1.15 < N_B < 1.5) \end{cases}$		
Quelle: ISO 6336-T1 [2006] (13-22)		

ID=82	MID=76	$\varepsilon_{\beta\_ref} = f(\varepsilon_{\beta})$
$\varepsilon_{\beta\_ref} = \begin{cases} \varepsilon_{\beta} & \text{if } \varepsilon_{\beta} \leq 1 \\ 1 & \text{otherwise} \end{cases}$		
Quelle: ISO 6336-T3[2006] (8.2)		

## 11 Anhang B

Projektdatei der Berechnung aus der Abbildung 90:

```
<?xml version="1.0" encoding="utf-8"?>
<Data>
  <Elements>
    <Element ID="20528282" Type="2" Func="Start" ParentName="" DisplayName="Start" X="225" Y="258">
      <Parameter Name="InputPorts"/>
    </Element>
    <Element ID="30643394" Type="1" Func="P4cceLVW40eP2" ParentName="" DisplayName="mt(mn,beta)"
X="553" Y="268">
      <Parameter Name="InputPorts">
        <Value>78780388</Value>
        <Value>74617787</Value>
      </Parameter>
      <Parameter Name="OutputPorts">
        <Value>89543185</Value>
      </Parameter>
    </Element>
    <Element ID="37189306" Type="3" Func="End" ParentName="" DisplayName="End" X="1035" Y="272">
      <Parameter Name="OutputPorts" />
    </Element>
  </Elements>
  <Ports>
    <Port ID="78780388" Type="23_17_01_01_MwkDwvh5w1r" Name="R_mn" Symbol="m^sn"
Description="Normalmodul" InitValue="0">
      <Parameter Name="Objects">
        <Value>30643394</Value>
      </Parameter>
    </Port>
    <Port ID="74617787" Type="23_17_01_01_Mun9j5BFPr1" Name="R_beta" Symbol="\x62"
Description="Schrägungswinkel" InitValue="0">
      <Parameter Name="Objects">
        <Value>30643394</Value>
      </Parameter>
    </Port>
    <Port ID="89543185" Type="23_17_01_01_MgpLc0rDMGk" Name="R_mt" Symbol="m^st"
Description="Stirnmodul" InitValue="0">
      <Parameter Name="Objects">
        <Value>30643394</Value>
      </Parameter>
    </Port>
  </Ports>
  <Connections>
    <Connection ID="C89723498" FirstObjID="20528282" SecondObjID="30643394" FirstConnectorID="0"
SecondConnectorID="1" />
    <Connection ID="C4836647" FirstObjID="30643394" SecondObjID="37189306" FirstConnectorID="0"
SecondConnectorID="1" />
  </Connections>
</Data>
```